
GLUCOSE DATA MONITORING SYSTEM AND DIABETES CONTROL TIPS

TRABAJO DE FIN DE GRADO



GRADO EN INGENIERÍA INFORMÁTICA

Tutor: Jose María Alvarez Rodríguez

Autor: Javier Villarejo Sánchez

21 DE JUNIO DE 2017
UNIVERSIDAD CARLOS III DE MADRID

Contenido

1.INTRODUCCIÓN	1
1.1-Motivación	1
1.2- Objetivos.....	2
2- ESTADO DEL ARTE	3
2.1- Objetivos y definición	3
2.2- La diabetes como enfermedad.....	4
2.2.1- El deporte y la diabetes	5
2.3- Computación móvil	8
2.4-Android.....	9
2.5- Los sistemas actuales	15
2.5.1- Sistemas de medición utilizados en la actualidad	15
2.6- Comparativa con aplicaciones similares	19
3-ANÁLISIS DEL SISTEMA	27
3.1- Especificación de requisitos software	27
3.1.1- Especificación de Requisitos Software (IEEE830-1998)	28
3.1.1.1- Propósito y alcance.....	28
3.1.1.2-Definiciones, acrónimos y abreviaturas	28
3.1.1.3- Perspectiva de producto.....	30
3.1.1.4- Funcionalidades del producto	30
3.1.1.4- Características de los usuarios.....	31
3.1.1.5-Restricciones.....	31
3.1.1.6-Suposiciones y dependencias	32
3.1.1.7- Evolución previsible del sistema	32
3.2- Requisitos específicos del sistema	33
3.2.1- Requisitos comunes a las interfaces	34
3.2.1.1-Interfaces de usuario.....	34
3.2.1.2-Interfaces de hardware	35
3.2.1.3-Interfaces de software.....	35
3.2.1.3-Interfaces de comunicación	36
3.3- Requisitos no funcionales	36
3.3.1- Requisitos de rendimiento.....	36
3.3.2- Requisitos de seguridad	37
3.3.3- Requisitos de fiabilidad	38
3.3.4- Requisitos de disponibilidad	38

3.3.5- Requisitos de mantenibilidad	39
3.3.6- Requisitos de portabilidad	39
3.4- Requisitos funcionales	40
3.4.1- Requisito funcional 1	40
3.4.2- Requisito funcional 2	41
3.4.3- Requisito funcional 3	42
3.4.4- Requisito funcional 4	43
3.4.5- Requisito funcional 5	44
3.4.6- Requisito funcional 6	45
3.4.7- Requisito funcional 7	46
3.4.8- Requisito funcional 8	47
3.4.9- Requisito funcional 9	48
3.4.10- Requisito funcional 10	48
3.5- Otros requisitos	49
3.5.1- Requisitos culturales y políticos.....	49
3.5.2- Requisitos legales.....	49
3.6-Casos de uso	50
3.6.1- Caso de uso 1: Creación de un usuario en el sistema	51
3.6.2-Caso de uso 2: Inserción de una cita médica	53
3.6.3-Caso de uso 3: Registrar una medida glucémica	54
3.6.4-Caso de uso 4: Mostrar detalles de una lista deslizable	56
3.6.5-Caso de uso 5: Eliminar registros y citas del sistema	58
3.6.6-Caso de uso 6: Mostrar gráficas de diferentes tipos	59
3.6.7-Caso de uso 7: Realizar recuperación de seguridad de los datos respaldados en Google Drive ..	61
3.6.8- Diagrama de casos de uso.....	63
4-DISEÑO DEL SISTEMA	65
4.1- Introducción al modelo 4+1	65
4.2- Vista lógica.....	66
4.3- Vista física o de despliegue	72
4.4- Vista de procesos.....	73
4.5- Mapa de pantallas	76
4.6-Entorno tecnológico.....	77
4.6.1- Desarrollo en Android.....	77
4.6.2- El entorno de programación de Android: Android Studio	78
4.7- Implementación.....	80
5- PRUEBAS DEL SISTEMA.....	85
5.1-Definición de casos de prueba	85
5.2-Trazabilidad del desarrollo del sistema	90
5.2.1-Matriz de trazabilidad: Requisitos/Casos de prueba	90

5.2.2-Matriz de trazabilidad: Casos de uso/Casos de prueba	90
6- MARCO LEGAL DEL SISTEMA	91
7- ENTORNO SOCIO ECONÓMICO.....	95
8- PRESUPUESTO Y PLANIFICACIÓN	101
8.1-Presupuesto.....	101
8.1.1-Costes directos:.....	101
8.1.2-Costes indirectos:.....	102
8.1.3- Beneficios.....	102
9- CONCLUSIONES Y TRABAJO FUTURO	107
10-ABSTRACT.....	109
11- REFERENCIAS.....	121

Índice de tablas

Tabla 1: Factores influyentes en la respuesta glucémica	7
Tabla 2: Causas y respuestas glucémicas en un paciente Diabético	7
Tabla 3: Evolución de Android y sus aspectos relevantes [11]	12
Tabla 4: Tabla resumen aplicaciones para el control de la glucemia	25
Tabla 5: Tabla de definiciones, acrónimos y abreviaturas	30
Tabla 6: Tabla de usuarios	31
Tabla 7: Plantilla de definición de requisitos	33
Tabla 8: Requisito 1	34
Tabla 9: Requisito 2	34
Tabla 10: Requisito 3	35
Tabla 11: Requisito 4	35
Tabla 12: Requisito 5	35
Tabla 13: Requisito 6	36
Tabla 14: Requisito 7	36
Tabla 15:Requisito 8	37
Tabla 16: Requisito 9	37
Tabla 17:Requisito 10	37
Tabla 18:Requisito 11	38
Tabla 19: Requisito 12	38
Tabla 20:Requisito 13	39
Tabla 21: Requisito 14	39
Tabla 22:Requisito 15	40
Tabla 23:Requisito 16	40
Tabla 24:Requisito 17	41
Tabla 25:Requisito 18	42
Tabla 26:Requisito 19	43

Tabla 27:Requisito 20	45
Tabla 28:Requisito 21	45
Tabla 29:Requisito 22	46
Tabla 30:Requisito 23	47
Tabla 31:Requisito 24	48
Tabla 32:Requisito 25	49
Tabla 33: Requisito 26	49
Tabla 34: Plantilla de tabla de casos de uso	51
Tabla 35: Caso de uso 1	53
Tabla 36: Caso de uso 2	54
Tabla 37: Caso de uso 3	56
Tabla 38: Caso de uso 4	57
Tabla 39: Caso de uso 5	59
Tabla 40:Caso de uso 6	61
Tabla 41: Caso de uso 7	63
Tabla 42: Caso de prueba 1	86
Tabla 43: Caso de prueba 2	86
Tabla 44: Caso de prueba 3	87
Tabla 45: Caso de prueba 4	88
Tabla 46: Caso de prueba 5	88
Tabla 47: Caso de prueba 6	88
Tabla 48: Caso de prueba 7	89
Tabla 49: Caso de prueba 8	89
Tabla 50: Matriz de trazabilidad Requisitos-Casos de prueba	90
Tabla 51: Matriz de trazabilidad Casos de uso-Requisitos	90
Tabla 52: Tabla de costes directos	103
Tabla 53: Tabla de costes indirectos	103
Tabla 54: Tabla de costes totales	103
Tabla 55: Tabla Gantt	104

Índice de ilustraciones

Ilustración 1: Definición de estado del arte [3].....	4
Ilustración 2: Datos del mercado móvil durante el año 2016 [9].....	8
Ilustración 3: Suscripciones móviles globales [9].....	8
Ilustración 4: Sistema de capas en Android [15].....	14
Ilustración 5: Sistema de capas Android [10]	15
Ilustración 6: Sistema de medición tradicional [16].....	16
Ilustración 7: Sensor de introducción continua [18]	17
Ilustración 8: Funcionamiento de una bomba de insulina [18].....	18
Ilustración 9: Interfaz de usuario de la aplicación "SocialDiabetes" [19]	19
Ilustración 10: Interfaz de usuario de la aplicación "Diario de diabetes mySgr" [21]	20
Ilustración 11: Interfaz de usuario de la aplicación "Diabetes Monitor" [22]	21
Ilustración 12: Ejemplo de carga de datos en el software "LibreView"	22
Ilustración 13: Informe de monitorización de glucemias (I) de la aplicación "LibreView".....	23
Ilustración 14: Informe de monitorización de glucemias (II) de la aplicación "LibreView"	23
Ilustración 15: Diagrama de casos de uso	63
Ilustración 16: Diagrama de componentes del sistema.....	69
Ilustración 17: Diagrama de clases del sistema.....	71
Ilustración 18: Diagrama de despliegue del sistema	72
Ilustración 19: Diagrama de secuencia (I)	74
Ilustración 20: Diagrama de secuencia (II)	75

<i>Ilustración 21: Mapa de pantallas del sistema</i>	<i>77</i>
<i>Ilustración 22: Distribución del desarrollo en Android por versión [31]</i>	<i>78</i>
<i>Ilustración 23: Android Studio [34].....</i>	<i>79</i>
<i>Ilustración 24: Fragmento de código 1</i>	<i>80</i>
<i>Ilustración 25: Fragmento de código 2</i>	<i>80</i>
<i>Ilustración 26: Fragmento de código 3</i>	<i>81</i>
<i>Ilustración 27: Fragmento de código 4</i>	<i>81</i>
<i>Ilustración 28: Fragmento de código 5</i>	<i>82</i>
<i>Ilustración 29: Fragmento de código 6</i>	<i>83</i>
<i>Ilustración 30: Fragmento de código 7.....</i>	<i>83</i>
<i>Ilustración 31: Los problemas de la obesidad a nivel mundial según la OMS [44]</i>	<i>96</i>
<i>Ilustración 32: El aumento progresivo de la diabetes como enfermedad [43]</i>	<i>97</i>
<i>Ilustración 33: Aumento del coste de la insulina Humalog [42]</i>	<i>98</i>
<i>Ilustración 34: Diagrama de Gantt</i>	<i>105</i>

1.Introducción

1.1-Motivación

Durante este documento, se va a tratar de aportar, una primera toma de contacto con los programas que han llevado a pretender realizar este proyecto de fin de grado, así como en algunos casos han matizado aspectos importantes que nuestro sistema pueda realizar. En él se mostrará cómo ha surgido la idea de realización, las motivaciones, así como algunos ejemplos prácticos o aplicaciones similares que, como se ha explicado antes, han sido llevadas a prueba y testadas durante algún tiempo para poder tener un juicio riguroso y específico de qué es lo que se quería realizar.

El objetivo principal es el desarrollo de un sistema capaz de registrar y monitorear de manera útil, aspectos básicos para el tratamiento de la Diabetes Mellitus Tipo 1 o también conocida como juvenil. Este sistema nos permitirá el registro y manipulación de los datos almacenados de una manera mucho más sencilla de la habitual en un sistema de escritorio, siendo posible su instalación en dispositivos móviles, con un conjunto de ventajas entre las que destaca la portabilidad para poder realizar dichos registros en tiempo real, al realizar nuestras mediciones. [1]

De esta manera comenzaremos explicando algunos aspectos personales que han podido influenciar para la realización de un sistema de esta índole. El hecho de que personalmente padezca esta enfermedad, ha sido y servido durante mi periodo formativo, para darme cuenta de que existe un vacío de aportaciones para este campo concreto de la ciencia, o al menos no del todo útiles en ciertos aspectos. Mi pasión por los dispositivos portables, la domótica o el auge de la conectividad entre dispositivos de los últimos tiempos, son solo algunos ejemplos que han hecho que me plantee cuál podría ser el resultado en caso de combinar, o alinear estos campos de conocimiento, formando uno único y común, que además pueda servir de ayuda a las personas que padezcan esta enfermedad.

Además, existe un factor de mayor peso si cabe, como es la posibilidad de concienciar a las personas sobre la gravedad que tiene en el mundo actual esta enfermedad, cuya importancia y repercusión, no alcanza las cotas de conocimiento y necesidad de desarrollo dignas para una enfermedad que involucra a tantas personas a nivel mundial.

Este afán por conseguir una unión entre dos ámbitos muy presentes en la vida de como digo, gran cantidad de personas, surge, de las nuevas tendencias tecnológicas que día a día tenemos más presentes en nuestra vida. Como bien sabemos, la vida está en continuo cambio, y la tecnología se va a encargar de modernizar y facilitar cada vez más todos los procesos que las personas hemos de realizar. Y es por esto por lo que la adaptación del entorno sanitario debería ser una fuente de motivación para estos avances, ya que mejoraría el principal y más importante ámbito de la vida

de una persona. Este aspecto se desarrollará más adelante en este documento, en un apartado especialmente dedicado a ello.

En este apartado, es donde surge otra motivación, que explicará otro de los puntos importantes de la realización de nuestro proyecto. Esta es la utilización de un sistema operativo móvil y no de escritorio. Este aspecto es algo francamente variable de un conjunto a otro de personas, pero la mayoría de personas a nivel mundial son acompañadas cada día por su teléfono móvil o tableta, por lo que la accesibilidad que este tipo de sistemas aporta, beneficiará muchos aspectos de nuestro desarrollo, como se ha comentado anteriormente. [2]

Asimismo, otro de los hechos más relevantes que me han llevado a pretender la realización de un proyecto de sistema así, se centra en el hecho de impulsar el “software libre”. No habiendo duda de que un problema que incumbe a tantas personas, no puede ser tratada únicamente por una sola, por lo que la ayuda entre miembros, es vital para que una aplicación pueda llegar a ser realmente de utilidad para un sector de la sociedad, y la mejora continua una de sus necesidades, llegando a ser posible una comunidad retroalimentada entre pacientes y desarrolladores.

1.2- Objetivos

Como se ha explicado en el apartado anterior, el objetivo durante este documento se centrará en la definición teórica y práctica de un sistema de registro y monitorización para el control de una enfermedad concreta, la Diabetes Mellitus Tipo 1 o juvenil.

Para la elaboración de este documento, se realizará una investigación sobre los temas principales que han motivado este desarrollo, organizándolos de manera estructurada. La primera de las partes se centra en realizar el estudio e investigación sobre los estados actuales de los diferentes factores comentados antes, situando al lector sobre la materia que va a desarrollarse.

Durante la segunda parte del desarrollo, nos centraremos en la definición formal de nuestro sistema, haciendo especial hincapié en lo que a desarrollo teórico se refiere, y tratando de plasmar y explicar todos aquellos aspectos relevantes, sobre los cuales se sustenta nuestro sistema software desarrollado. En este apartado se realizarán tareas de análisis, diseño, implementación y pruebas.

Por último, en la última sección del documento, se dispondrán los apartados del desarrollo que tienen que ver con los presupuestos, la planificación del proyecto, y las tareas realizadas durante el desarrollo e implementación de nuestro sistema.

Esto además incluirá un apartado que definirá el marco regulatorio y el entorno socioeconómico que ha involucrado a nuestro sistema, aspecto de vital importancia en un sistema de esta índole, pues trata aspectos sensibles legalmente, que pueden ocasionar problemas de privacidad, así como riesgos en la salud de las personas en caso de una utilización fraudulenta.

2- Estado del arte

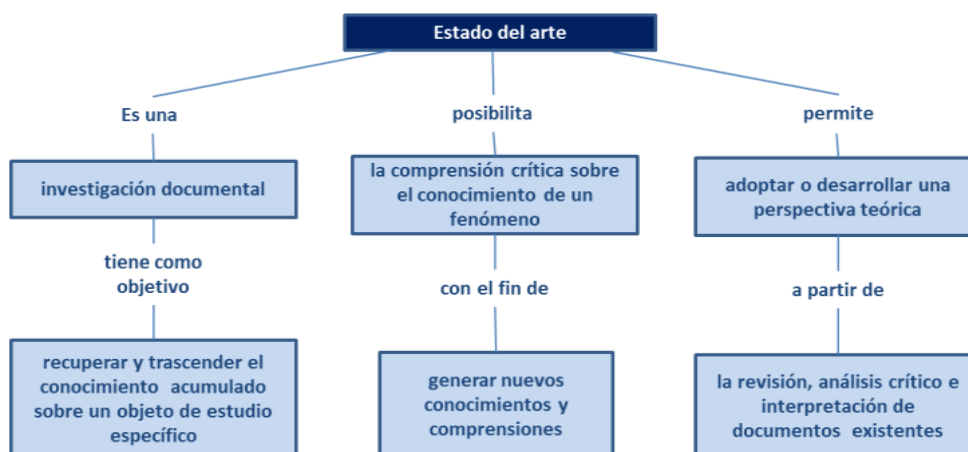
2.1- Objetivos y definición

La definición de estado del arte hace hincapié en un conjunto de procesos de investigación, así como un conjunto de pasos capaces de recuperar y describir, el estado actual de cualquier problema planteado de una manera documentada y organizada. Este estudio previo al planteamiento de una solución propia, permitirá crear una demanda de conocimiento, así como establecer comparaciones con conocimientos similares y que sigan una línea similar en su investigación y desarrollo, para obtener el mayor número posible de puntos de vista sobre nuestro problema planteado, y, por consiguiente, obtener una mejor implementación de nuestra solución.

De esta manera uno de los métodos más utilizados y, particularmente el que va a seguirse en este proyecto se centra en explorar temporalmente la evolución de todos los aspectos de nuestro problema. De esta manera se alcanzará con nuestra solución un nuevo paso en el desarrollo común entre todos los campos evaluados, y conseguiremos un avance potencial sobre los mismos. Se dice entonces que habremos alcanzado una punta de conocimiento, aportado un nuevo beneficio, mejora o novedad, sobre los campos estudiados, llegando de esta manera al punto más alto del desarrollo conocido hasta la fecha.

De esta manera, podemos introducir que el planteamiento de esta nueva puesta en situación sobre el estado de nuestro desarrollo, producirá en un futuro inmediato un nuevo conjunto de potenciales problemas que necesitarán de un nuevo estudio y nuevos planteamientos para poder superarse, por lo que nuestro estudio funcionará de manera retroalimentaría en el estado general del problema, y nunca de manera definitiva o única.

Así pues, la siguiente ilustración va a permitir entender de mejor manera cuales son los objetivos principales de este apartado del documento:



2.2- La diabetes como enfermedad

El primero de los puntos a investigar en la composición actual del estado del arte, ha de poner en situación a cualquier interesado en esta solución sobre qué es realmente la diabetes. Esta enfermedad es quizás una de las más presentes en el mundo actual, potenciada a su vez por el ritmo y modelo de vida que la sociedad desarrolla actualmente, pero a su vez es una enfermedad muy poco presente entre la población actual, siendo el desconocimiento y falta de información factores vitales para el mantenimiento de este motivo.

La diabetes mellitus, conocida simplemente como diabetes es un trastorno fisiológico cuyos factores o trastornos se centran en la falta de producción de insulina por parte del páncreas, o la imposibilidad de utilizar de manera adecuada las dosis que este órgano genera. El primero de los casos se definirá como *Diabetes Mellitus Tipo I o juvenil*, mientras que el segundo es tratado como *Diabetes Tipo II*. La insulina es un tipo de hormona generada en dicho órgano para poder permitir la absorción de hidratos de carbono, así como otras sustancias necesarias en cada una de las células de nuestro cuerpo para mantenernos vivos, y hacerlo de manera saludable. Esta incongruencia entre los niveles de insulina y los nutrientes obtenidos por parte de los alimentos consumidos, puede producir dos tipos de procesos en una persona que padezca dicha enfermedad, como son las hipo e hiperglucemias. Estas se centran en un nivel excesivo o insuficiente de insulina. El primero de ellos hará al cuerpo entrar en un estado de necesidad energética, llegando a producir sensaciones de debilidad y cansancio, llegando en casos extremos al coma diabético. Por lo que es totalmente necesario el consumo de hidratos de carbono en dichos casos. Por el contrario, el segundo de los casos se produce en la situación contraria, donde el déficit de insulina no permitirá al sujeto absorber la totalidad de los nutrientes, por lo que su ciclo sanguíneo portará un exceso de hidratos de carbono, siendo igual de peligroso, aunque los síntomas no sean tan específicos.

Para ambos tipos de diabetes existen un conjunto de factores de riesgo que pueden propiciar la aparición de dicha enfermedad. Vamos a establecer un pequeño resumen de cuáles son los más importantes:

Para la diabetes tipo I o juvenil [4], es decir aquella en la que el páncreas cancela en su totalidad la producción de insulina, no existen factores claros sobre el origen en un usuario de la enfermedad, pero se sospecha, y esto es algo que aún está en estado de investigación, que la genética o los antecedentes de algún familiar que haya padecido dicha enfermedad, pueden ser el motivo de la aparición de dicha enfermedad, en esta variante. Además, se valoran las posibilidades sobre la exposición a diferentes factores ambientales o infecciones virales, pueden también potenciar el desarrollo de dicha variación de la enfermedad.

Por el contrario, en el caso de padecer diabetes tipo II, sí que existen una serie de factores condicionantes a este tipo de enfermedad, a continuación, se muestran algunos de los posibles:

- Historial familiar de diabetes
- Sobrepeso excesivo, inactividad, mala alimentación
- Estados de embarazo (diabetes temporal), intolerancia a la glucosa (IGT)

Como puede apreciarse, la diabetes tipo dos no resulta un trastorno tan aleatorizado como pueda ser la diabetes tipo uno, pues se centra más en diferentes factores evaluables. En concreto son factores típicos de malos hábitos o en gran cantidad de casos de imposibilidad de realización, pues como veremos a continuación, la diabetes es una enfermedad muy normalizada en países subdesarrollados. [5]

Según la IDF, las últimas informaciones dictan que aproximadamente 415 millones de personas padecen de la enfermedad, ya sea de tipo uno o dos, y siendo esta una cifra que aumenta paulatinamente a un ritmo muy preocupante [5].

Este ritmo, está asociado a su vez a países con un índice PIB (Producto Interior Bruto) bajo, lo que significa que son países subdesarrollados o en vías de desarrollo, que disponen de menos medios para tener que ayudar a una mayor cantidad de población, estando esta además en continuo aumento.

Otro dato preocupante, se centra en que una de cada dos personas que padece esta enfermedad, están sin diagnosticar, es decir, no conocen los síntomas y desconocen el padecimiento de la enfermedad. Esto puede causar graves problemas en la salud de las personas, hasta el punto de llegar a la muerte. Esto se traduce en alrededor de tres millones de personas muertas durante el año 2015 [6].

Es por este conjunto de motivos por los que esta enfermedad, ha de ser tratada con otros ojos y desde otro punto de vista, dándole la importancia real que tiene, por su gravedad y por el conjunto de personas a las que implica, de todas las edades y de manera global.

2.2.1- El deporte y la diabetes

Durante el punto anterior, se ha tratado de explicar, a rasgos muy generales cuales son los aspectos más destacados de la diabetes como enfermedad. Pero aún existe un aspecto que tiene tal importancia en la vida de los pacientes con dicha enfermedad, que vamos a destacarle un apartado particular: este es el ejercicio físico.

Este es un aspecto del que, a pesar de los grandes avances a nivel de información y concienciación de los usuarios, no se ha conseguido argumentar con claridad cuáles son los beneficios sobre los pacientes. Según explica en un artículo el doctor Serafín Murillo, Asesor en nutrición y deporte para la fundación de la Diabetes e investigador del CIBERDEM (Hospital Clínic de Barcelona, 2013):

En la diabetes tipo 1 es donde actualmente existe más controversia. Parece ser que los estudios realizados no muestran con claridad cuál es el efecto del ejercicio sobre el control glucémico. Queda muy claro que una mayor actividad física provoca una reducción de las necesidades de insulina. Es decir, las personas más activas utilizan menos dosis de insulina que las sedentarias. Pero esto no significa que el ejercicio mejore los niveles de glucemia en personas con diabetes tipo 1. En este tipo de diabetes, el resultado de la glucemia depende del equilibrio entre insulina, alimentación y ejercicio. Si el deportista con diabetes no sabe regular acertadamente estos tres factores es probable que no mejore su control glucémico. Parece claro que es necesario un buen conocimiento de estos factores, insulina alimentación, y ejercicio, para conseguir una óptima regulación de la diabetes.

Con esta afirmación, se pone de manifiesto el temor que todo diabético Tipo 1 posee acerca de la realización de deporte, pues las causas pueden ser tanto positivas como, en caso de mala realización, negativas sobre la salud del paciente, siendo por tanto un apartado muy a tener en cuenta a la hora de controlar nuestros niveles de glucemia en sangre. Pero no cabe duda, por otro lado, que el ejercicio físico, siendo realizado de una manera controlada e inteligente, es capaz de mejorar una gran cantidad de aspectos, tanto fisiológicos como psicológicos, sobre la vida general de las personas. El ejercicio se traduce en un mayor flujo de sangre sobre nuestro sistema cardiovascular, una mayor oxigenación de los músculos y un favorecimiento de la nutrición de los mismos, debido a la mayor necesidad energética que requerirán los mismos, y que se obtendrá a través de la utilización los alimentos consumidos. [7]

Este concepto de oxigenación y nutrición muscular, tiene bastante que ver con el efecto que causa la insulina en los pacientes diabéticos. Como se ha comentado anteriormente, la insulina se comporta como si de una puerta se tratase entre las células musculares, y los nutrientes, más concretamente los hidratos de carbono, obtenidos en las ingestas de alimentos.

Es por tanto que no podemos dudar de que el mismo efecto que causa el ejercicio, es potenciado por la dosis de insulina que estemos consumiendo en el momento de la realización del ejercicio, de ahí su carácter positivo en el paciente, pero a la vez peligroso, pues su exceso podría suponer un riesgo evidente.

Además de este aspecto, cabe resaltar que existe otro caso en el que un diabético ha de tener especial cuidado sobre los niveles de glucemia antes de la realización de una actividad física. Esta se focaliza en el momento en el que los niveles de glucemia en sangre son excesivamente altos. En este caso, según dicta El grupo de trabajo de Diabetes y Ejercicio de la Sociedad Española de Diabetes (SED) [8]:

Si los niveles de insulina son demasiado bajos (situación de mal control metabólico y final del efecto de la insulina), el ejercicio físico provocará igualmente el aumento de producción de glucosa (liberación de hormonas contra reguladoras y niveles de insulina bajos), pero su utilización por el músculo estará disminuida (déficit de insulina), por lo que en vez de hipoglucemia aparecerá hiperglucemia, e incluso, en situación de

deficiencia marcada de insulina, cetosis. Por tanto, es importante monitorizar la glucemia previa al ejercicio y si ésta es mayor de 250 mg/dL, determinar la cetonuria. Si la glucemia es mayor de 300 mg/dL o bien si hay cetonuria, el ejercicio físico estará contraindicado hasta que se restablezca el control glucémico, para evitar una mayor descompensación.

Por tanto, a continuación, se introducirán un conjunto de tablas resumen [8], que permitirán observar las relaciones entre los factores influyentes de las glucemias en los pacientes diabéticos que practiquen deporte, y los efectos causados dependiendo de los mismos.

Factores influyentes en la respuesta glucémica

- **Ejercicio:**
 - Intensidad y duración.
 - Grado de entrenamiento.
- **Grado de control glucémico:**
 - >300 mg/dL y/o cetosis.
 - <300 mg/dL y ausencia de cetosis.
- **Tratamiento:**
 - Tipo y dosis de insulina.
 - Zona de inyección.
 - Tiempo desde la inyección e ingestión.
- **Nivel de insulinemia:**
 - Hipoinsulinemia.
 - Hiperinsulinemia.

Tabla 1: Factores influyentes en la respuesta glucémica

Causas y respuestas glucémicas

- **Se producirá un descenso de los niveles glucémicos si:**
 - Hiperinsulinemia durante el ejercicio.
 - Ejercicio de duración prolongada (>40-60 minutos) o intensidad alta.
 - Déficit de suplementos alimenticios.
- **La glucemia no cambia si:**
 - Ejercicio corto y poco intenso
 - Concentración de insulina normal
 - Suplementos alimenticios adecuados
- **La glucemia aumentará si:**
 - Hipoinsulinemia durante el ejercicio
 - Ejercicio muy violento
 - Suplementos alimenticios excesivos

Tabla 2: Causas y respuestas glucémicas en un paciente Diabético

2.3- Computación móvil

Las nuevas tecnologías cubren en la actualidad un porcentaje muy alto de nuestra vida cotidiana, así como un conjunto enorme de funciones y automatismos que nos facilitan el día a día. Esta tecnología aporta a diario un conjunto de servicios de los que cada vez somos más dependientes, siendo un nicho de mercado creciente, a un ritmo agigantado, y con una tendencia muy atractiva para las grandes empresas. Es por esto que tales incrementos, han generado nuevos campos de investigación y han permitido establecer nuevos objetivos o funcionalidades adaptables a la tecnología, así como una competencia creciente a nivel mundial entre marcas nuevas o provenientes de otros sectores también tecnológicos.

Vendor	Q1 2016 Shipment Volumes	Q1 2016 Market Share	Q1 2015 Shipment Volumes	Q1 2015 Market Share	Year-Over-Year Change
Samsung	81.9	24.5%	82.4	24.6%	-0.6%
Apple	51.2	15.3%	61.2	18.3%	-16.3%
Huawei	27.5	8.2%	17.4	5.2%	58.4%
Xiaomi	18.5	5.5%	7.3	2.2%	153.2%
Nokia	14.3	4.3%	6.4	1.9%	123.8%
Others	141.5	42.3%	159.8	47.8%	-11.4%
Total	334.9	100.0%	334.4	100.0%	0.2%

Source: IDC Worldwide Quarterly Mobile Phone Tracker, April 27, 2016.

Ilustración 2: Datos del mercado móvil durante el año 2016 [9]

Aquí se muestran algunos datos sobre el crecimiento global de los últimos años [9], donde se puede apreciar la tendencia masificada hacia la tecnología de las comunicaciones.

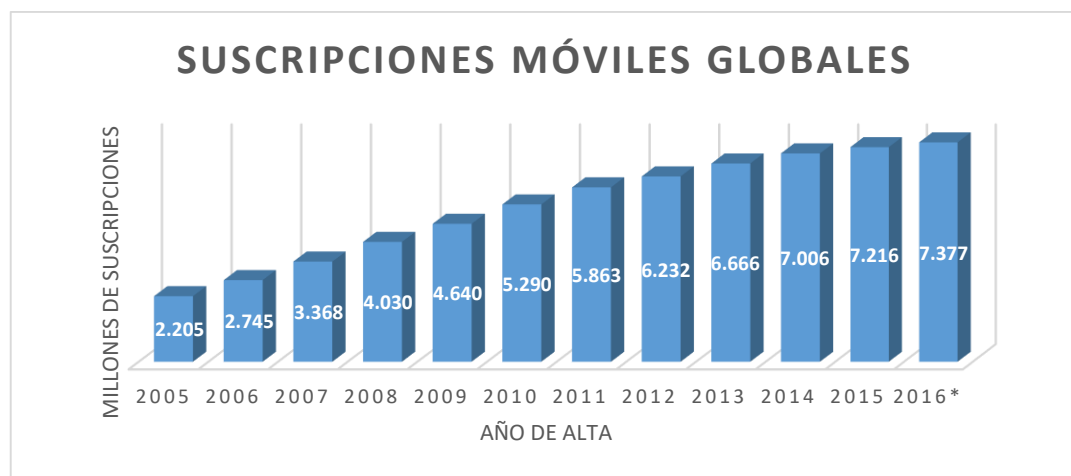


Ilustración 3: Suscripciones móviles globales [9]

Además de esto, existe un mercado de especial interés, en el que la ayuda de la tecnología es si cabe de mayor importancia: la medicina. Este aspecto es si cabe el de mayor importancia en la gran mayoría de personas, por lo que cualquier facilidad o avance que se produzca por la ayuda de las tecnologías, puede crear una gran repercusión en la vida general de las personas. Es por tanto que esta necesidad o demanda de avances, en conjunción con el gran crecimiento de investigación y desarrollo producido por el mercado móvil, crean la posibilidad de innovar de manera extraordinaria en ambos aspectos de la ciencia.

Y es precisamente en este campo, en el que cobra mayor sentido el desarrollo del planteamiento que precede, pues como ya se ha introducido, existe una necesidad intrínseca entre los pacientes diabéticos, y la necesidad de incorporar nuevas herramientas y mejoras en los dispositivos utilizados para el correcto control de dicha enfermedad crónica. Por ello, de aquí en adelante, van a introducirse un conjunto de sistemas, tanto software, como los tradicionales sistemas de medición hardware, para entender cómo se aplicará en este entramado nuestro sistema propuesto.

Con este propósito, además nos dispondremos a analizar qué herramientas serán las utilizadas para el desarrollo de nuestro sistema, además de la necesidad de ser un sistema accesible y manipulable por un conjunto amplio de usuarios.

2.4-Android

“Android es un sistema operativo inicialmente pensado para teléfonos móviles, al igual que iOS, Symbian y Blackberry OS. Lo que lo hace diferente es que está basado en Linux, un núcleo de sistema operativo libre, gratuito y multiplataforma.”

[10]

Esta es una de las definiciones que podríamos encontrar en revistas especializadas sobre el sistema operativo móvil, desarrollado y mantenido en la actualidad, por la multinacional Google.

Pero el sistema operativo del gran buscador tiene una gran cantidad de afiliaciones y conceptos, por lo que trataremos de definir el mismo, de manera más concreta y precisa a lo largo de la historia.

La historia de Android comienza en el año 2004, cuando un conjunto de desarrolladores, llevan a cabo lo que probablemente sería uno de los negocios más reputados, y con un mayor peso en la industria mundial. **Andy Rubin** (co-fundador de Danger), **Rich Miner** (co-fundador de Wildfire Communications, Inc.), **Nick Sears** (alguna vez VP en T-Mobile), y **Chris White** (quien encabezó el diseño y el desarrollo de la interfaz en WebTV), fueron los co-fundadores de este sistema que apenas tuvo repercusión en sus inicios, pero que vieron como en Julio del año 2005, todo su trabajo se veía recompensado con la compra por parte de Google. La gran multinacional adoptó a la pequeña compañía Android Inc. fundada en Palo Alto, California para entrar de lleno en el terreno de los dispositivos móviles.

A partir de aquí el equipo de Andy Rubin comienza un proyecto de desarrollo en la sombra, de la mano de la gran compañía de la que poca información se tiene, pero es en el año 2007, cuando Google decide dar un paso de gran importancia en el futuro del negocio, formando parte de la **Open Handset Alliance** [11]. Esta nueva comunidad formada por diversas marcas tecnológicas, de dispositivos móviles, y de fabricantes de componentes, que estandarizó dicho mercado y estableció sus primeros pasos.

De esta manera es en el año 2008 cuando tras varios acuerdos con una gran cantidad de marcas de reputación como **PacketVideo**, **“ARM” Holdings**, **Atheros Communications**, **Asustek**, **Garmin**, **Softbank**, **Sony Ericsson**, **Toshiba**, **Vodafone** y **ZTE**, sale a la venta el primer terminal denominado “Smartphone” con el sistema operativo de la gran G corriendo en su interior, generado sobre un kernel Linux, y con su SDK liberado, para que los desarrolladores fueran capaces de alimentar el sistema operativo con nuevas funcionalidades, y aplicaciones.

EVOLUCIÓN HISTÓRICA DEL SISTEMA OPERATIVO ANDROID

Versión de Android	Número de versión	Año de lanzamiento	Principales actualizaciones
Android 1.0	1.0	Septiembre de 2008	Integración con Google Play Services
			Navegador web, integración con web HTML y XHTML, y trabajo con ventanas
			Integración de Andorid Market Implementación de módulos de multitarea, Wi-Fi, Bluetooth y mensajería
Cupcake	1.5	Abril de 2009	Mejoras en la interacción con la cámara
			Mejoras en el GPS y localización Implementación del teclado virtual
Donut	1.6	Septiembre de 2009	Implementación de las búsquedas por voz
			Creación de las galerías fotográficas y mejora en el rendimiento de la grabación de vídeo

					Creación de indicadores de batería
					Soporte CDMA
					Multilenguaje en búsquedas por voz
Eclair	2.0	Octubre 2009	de		Sincronización multicuenta
					Soporte para Microsoft Exchange
					Bluetooth 2.1
					Navegador web con interfaz HTML5
					Implementaciones de calendario
Froyo	2.2	Mayo 2010	de		Implementación de los widgets en pantalla
					Mejoras en el soporte Exchange
					Soporte Hotspot
					Traducción del teclado a diferentes idiomas
					Introducción de Adobe Flash 10.1
Gingerbeard	2.3	Diciembre de 2010	de		Mejoras en la interfaz para aumentar la rapidez y simplicidad en los gestos
					Mejoras en el teclado y la introducción de texto
					Implementación y adaptación de la función de copiar/pegar
					Implementación del NFC
Honeycomb	3.0	Mayo 2011	de		Adaptación del sistema para tabletas y dispositivos con mayor tamaño de pantalla
					Mejoras en la multitarea, las notificaciones, personalización y widgets
					Envío de contenido a través de bluetooth
Ice Cream Sandwich	4.0	Octubre 2011	de		Nuevo formato de fuente <i>Roboto</i>
					Nuevos medios de desbloqueo
					Mejoras en la seguridad para el envío de datos (Android Beam)

				Introducción de las capturas de pantalla, creación de carpetas de aplicaciones
				Implementación de gestos para descartar notificaciones
				Implementación del Wi-Fi Direct y compartición de internet
Jelly Bean	4.1	Julio de 2012		Mejoras en la rapidez, fluidez y respuesta de la interfaz
				Widgets escalables en pantalla
				Dictador de voz offline Google Now
				Mejoras en la seguridad de Android Beam
				Mejoras en la calidad y rapidez de las actualizaciones de las aplicaciones
KitKat	4.4	Septiembre de 2013		Soporte MAP bluetooth
				Implementación de nuevas transiciones de la interfaz
				Impresión wireless
				Mejoras en la multitarea y optimizaciones de la memoria
Lollipop	5.0	Marzo 2015	de	Cambio en la plataforma de ejecución, pasando de Dalvik a ART
				Mejoras en el "recolector de basura"
				Llamadas en alta definición entre dispositivos con 4G disponible
Marshmallow	6.0	Mayo 2015	de	Incorporación de permisos en tiempo de ejecución
				Implementación de soporte para huellas digitales de manera nativa
Nougat	7.0	Julio de 2016		Incorporación de la API de Vulkan para la mejora de los gráficos 3D
				Implementación del modo DOZE de ahorro de energía

Tabla 3: Evolución de Android y sus aspectos relevantes [12]

A partir de este momento, Android como sistema operativo no ha parado de crecer. Introduciendo desde su versión inicial (2.6), hasta la más reciente conocida y aún en desarrollo, Android 8.0 o también denominada como Android O [12]. Pero estas mejoras no se han producido únicamente a un nivel visual. Android engloba muchos otros aspectos que van a tratarse a continuación:

-La **administración de la CPU** [13] es uno de los aspectos más relevantes de este particular sistema operativo. Se divide en cuatro partes básicas: El **kernel**, el **Middlewere**, las aplicaciones y la interfaz de usuario.

La primera de ellas podría definirse como el enlace entre nuestro hardware del dispositivo y el sistema de drivers. Estos drivers se encargarán de la gestión de memoria, la velocidad de los procesadores, el uso de sensores, etc.

La segunda, el **Middlewere** podría definirse como un conjunto de módulos capaces de adaptar las aplicaciones a dispositivos móviles, mediante un conjunto de **códecs**, servicios de mensajería, seguridad móvil, etc.

Las aplicaciones se centran en un entorno propio generado por un gestor de aplicaciones, ejecutadas independientemente con código programable para permitir el desarrollo independiente, pero ejecutarse como si se tratara de una máquina virtual.

La interfaz de usuario que se centra en la interacción del usuario con la propia máquina, y con cada una de las aplicaciones que esta ejecuta. Este ha sido uno de los puntos que más ha mejorado, para posibilitar al usuario, no solo la interacción táctil, sino la realización de dicha interacción de una manera más intuitiva y más rápida para realizar cualquier acción.

-La **gestión de la memoria** es otro de los puntos importantes en Android. Este es uno de los aspectos que más ha mejorado en las últimas versiones del sistema, permitiendo mantener en segundo plano las aplicaciones a modo de multiproceso, sin tener que influir en el rendimiento. Esto se produce debido a que Android hace correr cada una de sus aplicaciones como un proceso independiente, y a su vez posee un gestor propio capaz de cerrar aplicaciones que ralenticen otros procesos o la propia fluidez del sistema.

- La **seguridad** es sin ninguna duda uno de los motivos por los cuales podríamos decantarnos por un sistema operativo u otro. A medida que avanzan las tecnologías, también lo hacen los métodos para invadir la privacidad de las personas y violar sus derechos. Es por esto que Android [14] ha invertido una gran cantidad de tiempo de desarrollo en este aspecto, teniendo unas consignas claras:
- El hecho de que Android se haya desarrollado como un sistema operativo abierto u “**open source**”, hace necesaria la integración en el mismo de un sistema de seguridad robusto. Esto se consigue a través del desarrollo de una seguridad multi-capa, capaz de mantenerlo como una plataforma abierta y segura.

- Una de las partes más importantes de Android, es el enfoque hacia los desarrolladores, permitiéndoles crear software de manera libre y a su antojo. Esto puede hacer que se puedan pasar por alto sistemas atacantes contra la plataforma o los usuarios consumidores de la misma. Por este motivo Android implementa un conjunto de herramientas (Google Play, Google Play Services, etc.) capaces de reconocer vulnerabilidades y ayudar al desarrollador a solventarlas. Esto funcionará como una biblioteca de seguridad capaz de reconocer amenazas potenciales y evitarlas.
- Por otro lado, Android es un sistema completamente orientado al mercado móvil [13], y como sabemos este mercado está en gran medida enfocado a los usuarios. Por ello, Android en sus últimas actualizaciones ha hecho hincapié en la creación de un sistema complejo de permisos, que permita al usuario definir qué y qué tipo de permisos acepta. Este concepto está claramente enfocado a la evitación de malware (software malintencionado), capaz de recuperar datos sobre el usuario sin su permiso, o realizar acciones maliciosas.

Para todos estos motivos, Google ha tratado de establecer una política de actualizaciones continua, lanzando en cortos espacios de tiempo, aproximadamente cada mes, nuevas actualizaciones que corrijan errores y vulnerabilidades sobre su sistema operativo.



Ilustración 4: Sistema de capas en Android [15]

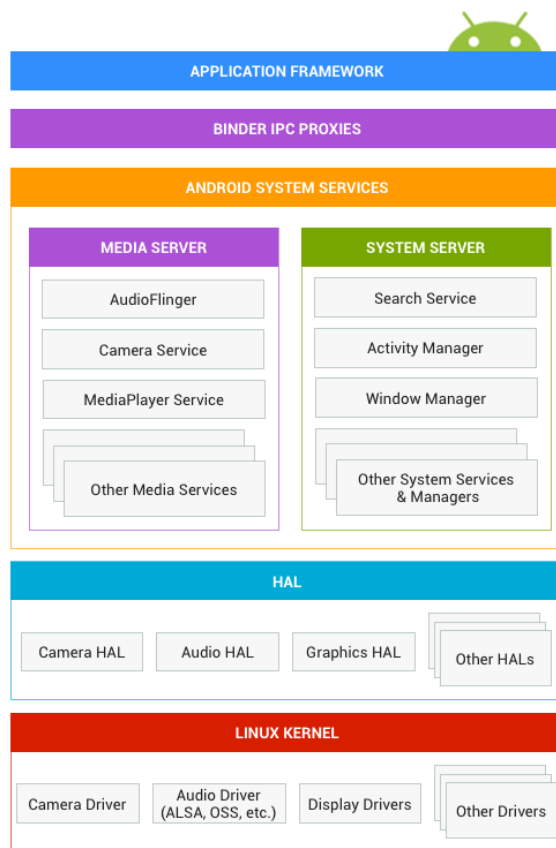


Ilustración 5: Sistema de capas Android [10]

2.5- Los sistemas actuales

Durante los dos apartados anteriores, se han introducido dos conceptos que claramente van a marcar la tendencia seguida durante todo nuestro proyecto. Pero aún faltan por definir cuáles son claramente las competencias que va a tener nuestro sistema. Con competencias definimos todos los dispositivos que ejecuten una labor similar, o potencialmente parecida, a lo que nuestro sistema implementará, derivando así en un estudio completo y eficaz.

2.5.1- Sistemas de medición utilizados en la actualidad

Esta clase de sistemas son los implementados por un dispositivo de medición capaz de aportar al usuario los valores de su glucemia en sangre, para de esta manera, ser capaz de controlar dicha medida y mantenerla en unos valores estables, mediante la acción de la insulina. Este conjunto de dispositivos es vital para la vida de un paciente con este tipo de enfermedad, pues es lo única que le permitirá medir o controlar los niveles de los que va a depender el estado de su enfermedad.

Es por ello que la tecnología ha tomado una gran acción en este campo, y cada vez de mayor manera, se están desarrollando medidores más reducidos en tamaño, con un mayor aporte de funcionalidades y valores de interés, tales como la influencia de unidades de insulina al usuario con respecto a una dieta, o la sugerencia y autocorrección de unidades de insulina, a través de sistemas personalizados por el

propio paciente en conjunción con su médico. Esto lo hacen cada vez más sencillos de utilizar, y cada vez más económicos, sobre todo por la introducción de nuevos métodos de medición, como los **dispositivos de introducción continua**.

Esta diferencia introducida es de gran valor en la actualidad, y se ha producido un desarrollo muy importante en este campo. Los avances más destacados giran en torno a dos conceptos diferentes. El primero se centra en la medición de glucemias sanguíneas en cada paciente, y la precisión y tardanza de dicha medición. En este aspecto la tecnología más tradicional se centra en realizar un pinchazo por cada medición que se desea realizar, e introduciendo una pequeña cantidad de sangre en una tira reactiva, conectada al dispositivo, obtener nuestro nivel de glucemia en tiempo real.

Los dispositivos más tradicionales, como ya se ha introducido anteriormente, consistente en la medición según los criterios del paciente, en los horarios que el establezca y en los momentos que desee. Estos se producen mediante un pequeño pinchazo en las yemas de los dedos, produciendo una pequeña hemorragia sanguínea, que permitirá introducir en la tira reactiva una dosis de sangre, para que este mida la glucemia en sangre (mg/dl) que circula por el corriente sanguíneo del paciente en ese instante.

El problema con este tipo de dispositivos, a pesar de que son los más extendidos por su simplicidad y coste reducido, es que las manos deben estar totalmente limpias, pues cualquier alimento o sustancia puede alterar en gran medida los resultados ofrecidos por el sistema. Como se ha comentado, este tipo de dispositivos cada vez son más pequeños, portables, e incluso existen algunos que no dependen de tiras reactivas, sino que miden directamente el nivel de glucemia, pero el coste en consecuencia es mayor. Otro de los inconvenientes que posiblemente tenga este sistema de medición, es el recambio continuo de las tiras reactivas, ya que la utilización constante en el día a día, hará que tengas una dependencia para conseguir dicho material, que es específico por marca y por dispositivo.



Ilustración 6: Sistema de medición tradicional [16]

Pero la tecnología avanza y se ha introducido un nuevo concepto, que hace al paciente mucho más independiente del medidor, a la par que limita el número de pinchazos necesarios a lo largo del día. Estos son los dispositivos de introducción o

medición continua. Su funcionamiento se centra en la aplicación sobre la piel de un parche de pequeño tamaño, vinculado mediante NFC (Near Field Communication) [17] a un dispositivo.

Este será encargado de recoger valores de glucemia durante espacios temporales muy cortos, para de esta manera obtener un valor muy necesario, que es la tendencia que la glucemia tiene en el paciente. Esto puede ser de gran utilidad, por ejemplo, antes de la realización de prácticas deportivas, pues el paciente conocerá si sus niveles de insulina tienen un exceso o defecto, antes de comenzar dicha actividad y poder sufrir un riesgo de híper o hipoglucemia.

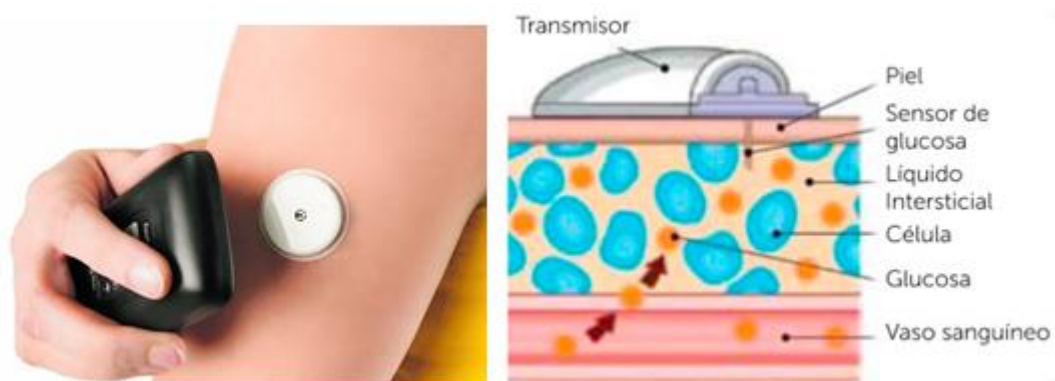


Ilustración 7: Sensor de introducción continua [18]

Por otro lado, existe otro concepto que está en desarrollo [18], como es la introducción de insulina en el paciente. Este aspecto tiene un factor si cabe más importante, como es la prudencia a la hora de recomendar ciertas dosis. El defecto o exceso en dicha estimación por parte de cualquier dispositivo, podrá generar valores de la glucemia del paciente indeseados, corriendo el riesgo de padecer descensos hasta niveles peligrosos. Es por tanto que el paciente nunca debe confiar el control total de las unidades suministradas al dispositivo.

Con respecto al funcionamiento, esta clase de dispositivos combinan en cierta parte los explicados anteriormente, de introducción continua, para conocer el nivel de glucemia del paciente y mediante una bomba independiente, introducir pequeñas dosis de insulina en los momentos precisos, para de esta manera mantener los niveles estables. Funcionan a modo de petaca, ajustable en un pantalón o cinturón, y deben ir continuamente pinchados en la piel, mediante una cánula, para poder funcionar. Este es el principal defecto de dichos dispositivos, la dependencia de estar continuamente clavados en la piel.

Cabe destacar que el tipo tradicional de dispositivos, ha evolucionado también de ser jeringas en las cuales se debía medir e introducir la cantidad de insulina, que se encontraba en botes independientes, a la introducción de insulina almacenada en

plumas, que contienen una cantidad de insulina propia, y con la ayuda de un selector de unidades nos permitirá seleccionar la cantidad a inyectar.

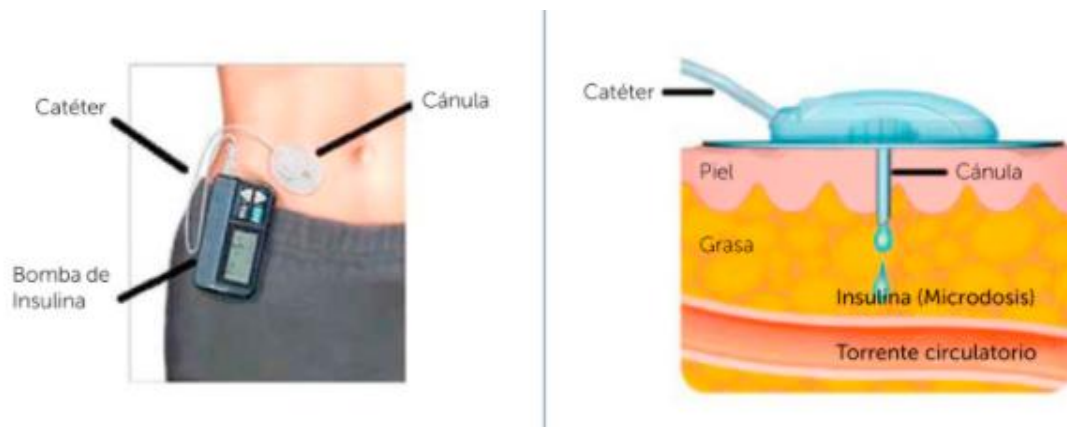


Ilustración 8: Funcionamiento de una bomba de insulina [18]

Pero la tecnología actual nos permite realizar una diversidad de cosas mucho mayores. Por eso se ha decidido incluir un nuevo sub apartado que introduzca una clase de sistemas no mencionados hasta ahora El auge de los dispositivos portables, ya sean móviles, tabletas o dispositivos “**wereables**” o vestibles, abren un nuevo campo de interés para esta clase de sistemas de control y manipulación para los diabéticos. Este conjunto, dada su repercusión, hace que cada día salgan a producción gran cantidad de aplicaciones útiles para diferentes apartados de nuestra vida rutinaria, que nos premeditaran obtener nuevas funcionalidades y facilidades a la hora de desarrollar nuestra vida. Es por esto que en este apartado se quieren introducir las aplicaciones capaces de controlar la diabetes. Estas aplicaciones, que poseen desarrolladores muy variados, desde grandes empresas del sector, hasta desarrolladores propios, cuentan con una aceptación no demasiado amplia, pues los métodos tradicionales de gestión transmiten más fiabilidad al usuario, en una recopilación de datos tan delicada.

Estas aplicaciones, a continuación, estudiadas y probadas durante algún tiempo, nos permitirán ingresar medidas de glucemia, valores de inyección de insulina, ingesta de alimentos y ejercicio, entre otras. Pero no funcionan únicamente a modo de diario o almacén de datos. Estas aplicaciones pueden monitorear dicha información introducida por el usuario, y generar información útil para el mismo. Esto se traduce en generación de gráficas, avisos por valores atípicos, recomendaciones de alimentación o necesidad de insulina, etc. En un dispositivo personal.

2.6- Comparativa con aplicaciones similares

En este apartado vamos a tratar de realizar una comparación entre nuestra idea, y las posibles aplicaciones que existen actualmente en el mercado.

La primera se trata de “**SocialDiabetes**” [19], una aplicación muy bien diseñada, que respeta los patrones de diseño de **Material Design** [20], ya que posee una versión de IOS, heredando de esta manera el buen hacer de estas aplicaciones. Con lo que respecta a los aspectos médicos, la aplicación tiene una gran cantidad de opciones, y nos permite adaptar nuestro tipo de diabetes, dependiendo de las respuestas ofrecidas a unas preguntas tipo. Esto es de gran utilidad a la hora de categorizar de alguna manera a los diferentes perfiles de la enfermedad que puedan existir, y permitirá hacer un mejor trabajo a la hora de realizar alguna recomendación.

La aplicación funciona introduciendo un conjunto de registros, con datos como la fecha, hora y nivel de glucemia, así como la cantidad de carbohidratos consumida por el usuario. A partir de dichos valores, se generarán gráficas orientativas y recomendaciones sobre valores conflictivos para la salud del paciente, generados a través de dichos registros.



Otra buena opción, muy valorada por los usuarios en la tienda de aplicaciones, se trata de “**Diario de diabetes mySgr**” [21]. Esta aplicación, tras su uso, peca quizás de un peor diseño que la anteriormente citada, pero a su vez es más sencilla, intuitiva y rápida, gracias a tener en un vistazo rápido, todos los datos importantes.

Posee demasiadas opciones de “personalización” para la realización de registros de glucemia, lo que ralentiza el proceso en el que se guardan dichas consultas de medidas. Esto es algo que puede no contentar a muchos usuarios, o puede resultar

complicado para personas adultas. Además de esto su interfaz, es un tanto más técnica, utilizando demasiados menús, siendo estos además mas complejos, por lo que requiere de un conocimiento más avanzado para poder utilizarse, así como una curva de aprendizaje mayor para aprender a sacarle el mayor provecho.

Sin embargo, posee un apartado muy interesante para poder obtener datos de diferentes medidores glucémicos, lo que hará que sea un plus en el control, pues dichos monitores están específicamente diseñados para ese uso, y recopilan mayor cantidad de datos, con mayor precisión.

A la hora de registrar una medida, como ya se ha producido a la hora de navegar a través de los menús de la aplicación, es necesaria la introducción de demasiados parámetros, aunque aporta gran cantidad de información si el usuario no tiene problemas en tomarse algo más de tiempo en este apartado.

Como en el caso anterior, genera gráficas temporales, que permiten visualizar de manera más general nuestros niveles de glucemia, añadir fotos de los alimentos consumidos, o conocer la media de los niveles del día.

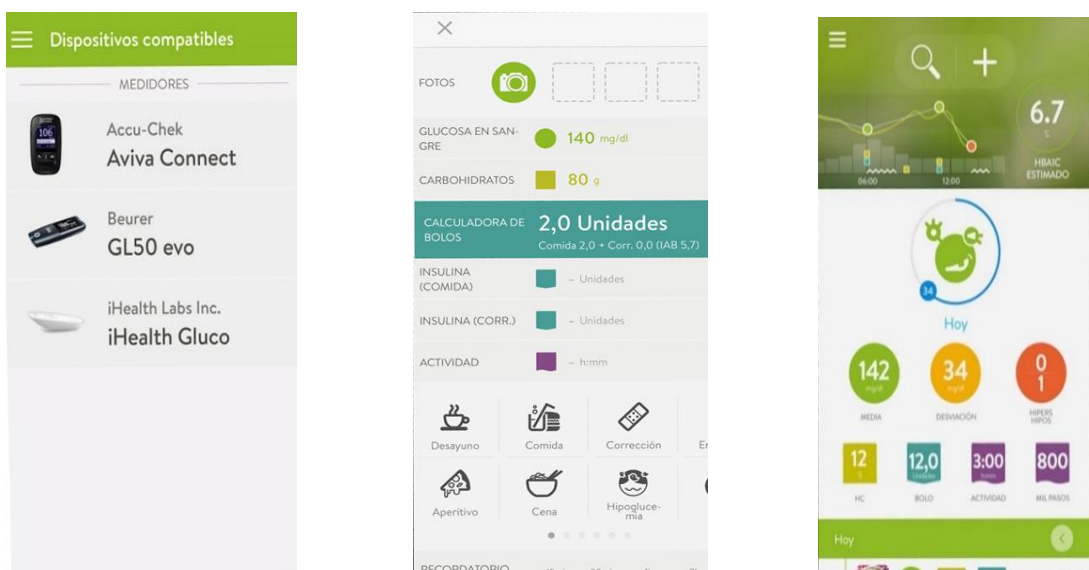


Ilustración 10: Interfaz de usuario de la aplicación “Diario de diabetes mySgr” [21]

La tercera de las opciones que en mi caso he podido probar, es la aplicación “**Diabetes Monitor**” [22]. Esta aplicación tiene una muy buena puntuación en la tienda de aplicaciones de Google pues tiene una gran característica, que es la adición de ejercicios físicos personalizados, algo que se verá más adelante, tiene una gran repercusión. Posee las disponibilidades básicas del resto de aplicaciones, como son el registro de mediciones y de consumo de alimentos, el calendario, y gráfico de estudio de medidas.

Sin embargo, la interfaz a pesar de respetar los patrones actuales de diseño, esta no está bien distribuida. La navegación entre pestañas está demasiado presente, pues los datos mostrados en las mismas son demasiado precisos, incomodando esa visión generalizada de valores que permita que el paciente se abstraiga y obtenga los datos de interés en un vistazo. Esto además se ve lastrado en el uso de la aplicación, pues su curva de aprendizaje es también muy amplia, siendo necesario cierto tiempo para poder aprender a utilizarla.

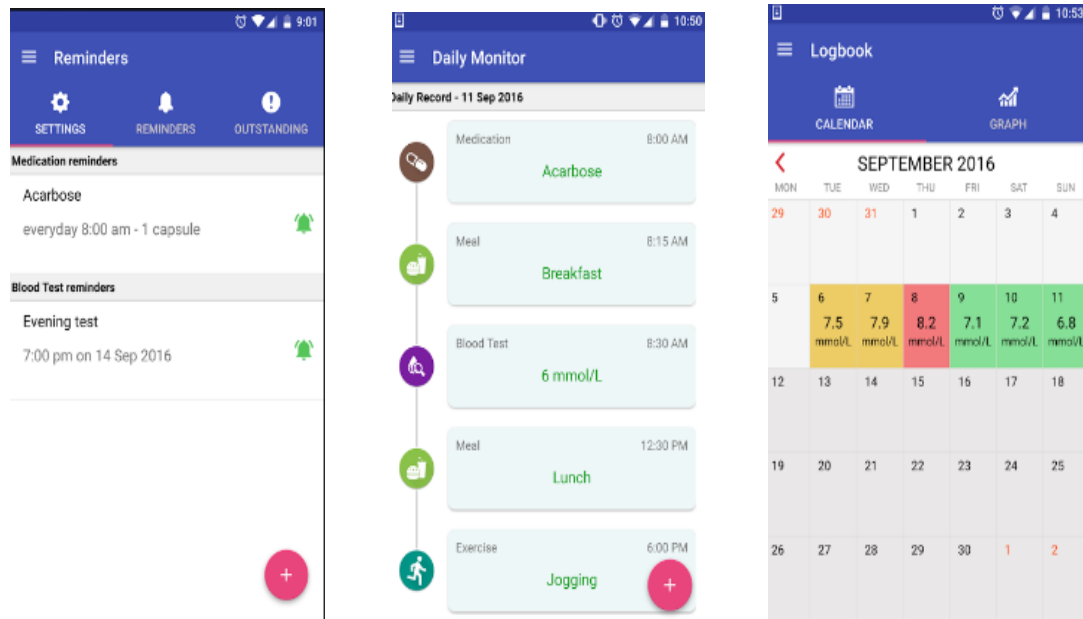


Ilustración 11: Interfaz de usuario de la aplicación "Diabetes Monitor" [22]

Existen en este campo de los sistemas software, un conjunto que no se podrían introducir en el mismo grupo que las tres aplicaciones anteriormente citadas. Estas aplicaciones son las desarrolladas por las empresas médicas para cada uno de sus dispositivos. En este caso, únicamente hablaremos de la aplicación propia proporcionada por **Abbot**.

Esta se llama concretamente "**LibreView**" [23], y es una aplicación muy bien desarrollada, no solo por el entorno, el cual se sale un poco de los tres ejemplos anteriores, comportándose como una aplicación web a la hora de cargar los datos, y como una aplicación disponible para dispositivos móviles si simplemente queremos consultarlos, ofreciendo una facilidad de uso para al usuario magnificada.

La diferencia entre este tipo de aplicaciones y las citadas anteriormente, es que estas están directamente vinculadas con nuestros dispositivos de medición, ya sea de medición tradicional, continua, o bomba de insulina. Eso favorece a la integración propia de un dispositivo conocido, y una aplicación por tanto diseñada concretamente para este, lo que hace que no deba adaptarse a diferencias de

formatos de datos, tipos, o drivers dentro de la misma. Sin embargo, el único inconveniente es que, para realizar la carga de las medidas, que como se explicará más adelante, es de manera automática, necesitaremos de un dispositivo con un sistema operativo de escritorio.

Otro de los inconvenientes que tiene esta aplicación, con respecto a las anteriores, es las limitaciones que tiene a la hora de introducir medidas, ya que la misma no permite la edición de los datos obtenidos, y deberemos modificarlos en el dispositivo, algo que aún está en vías de desarrollo, debido a las limitaciones hardware que estos tienen. Es por esto que, en caso de registrar una medición errónea, esta al realizar la carga se almacenará directamente en nuestro sistema, sin dar la posibilidad de modificarla a posteriori. Esto hace que definamos esta clase de sistemas, como sistemas únicamente de monitorización de medidas, y no de registro.

Tras conectarlas a nuestro ordenador, nos proporcionarán datos de gran interés como tendencias, porcentajes de distribución de medidas, glucemias promedio, o tiempo entre rango de medidas.

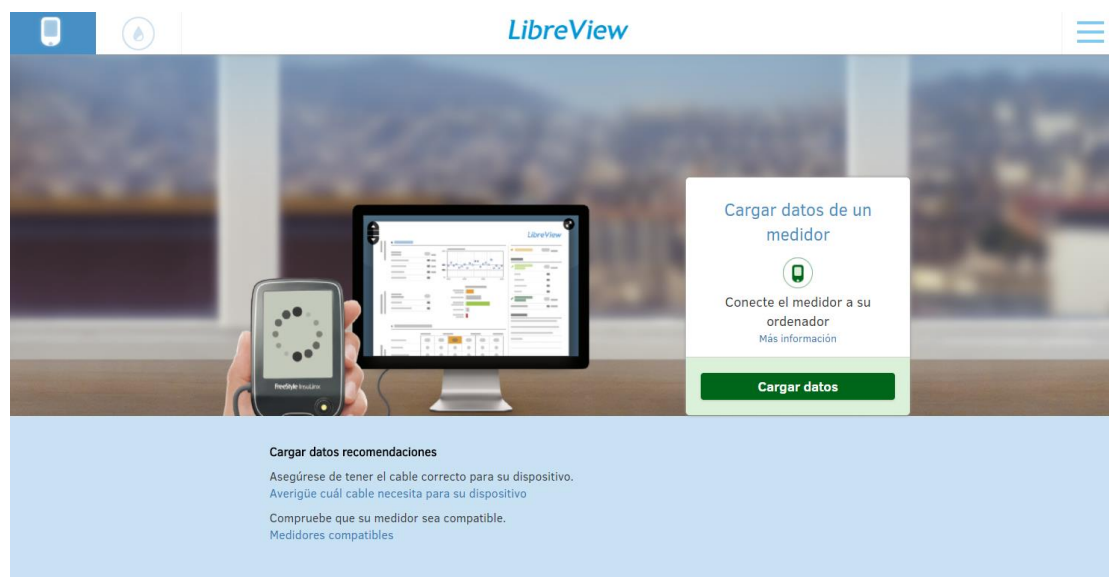


Ilustración 12: Ejemplo de carga de datos en el software "LibreView"

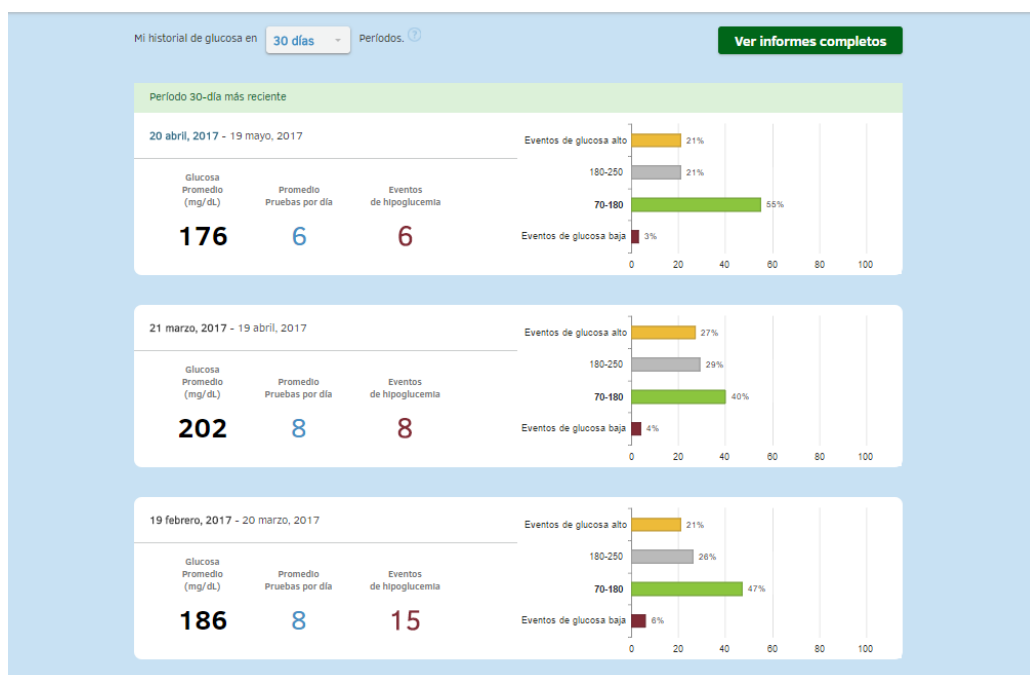


Ilustración 13: Informe de monitorización de glucemias (I) de la aplicación “LibreView”

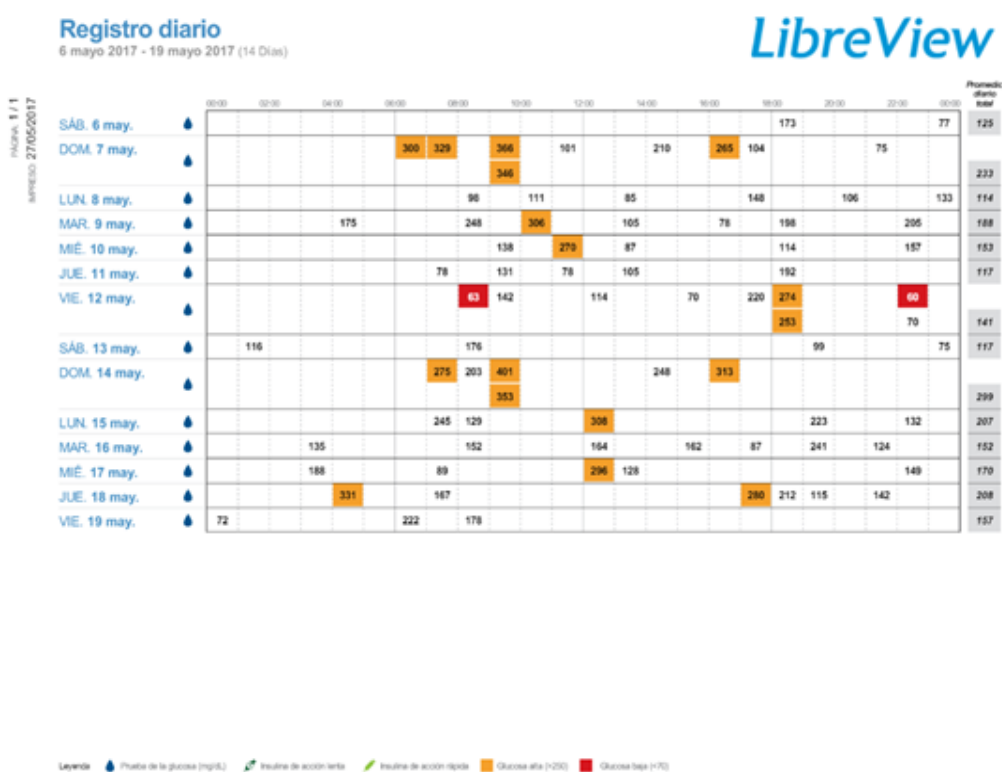


Ilustración 14: Informe de monitorización de glucemias (II) de la aplicación “LibreView”

**Capturas obtenidas de la propia aplicación durante su uso.

De esta manera, y tras el breve análisis y explicación de cada uno de los sistemas que han podido probarse de primera mano, definiremos en la siguiente tabla, a modo de resumen y comparativa, cada una de las mismas, para de esta manera conocer cuál de ellas hemos de utilizar, o al menos, cuáles han sido los aspectos más gratificantes en cada una de ellas.

En esta se valorarán aspectos como el diseño de la interfaz, las navegaciones entre menús, calificando estas según su grado de rapidez e intuición para el usuario, la monitorización mediante gráficas, que es un aspecto muy valorado para la abstracción a diferentes medidas en el tratamiento de la enfermedad.

Además, se valorará la sencillez para registrar nuevos valores de medidas, que permitirá al usuario introducir valores, editarlos, o inclusive introducir otros aspectos como el deporte realizado, o la ingesta de alimentos.

Por último, se categoriza la aplicación según sea aplicación móvil, de escritorio, o si dispone de la posibilidad de conectar con dispositivos externos para obtener datos de los registros de glucemia.

Aplicación	Diseño e interfaz	Navegación entre menús	Monitorización gráfica	Sencillez en el registro de medidas	App móvil	App escritorio	Conexión con dispositivos externos
SocialDiabetes	Buen diseño y disposición respetando los patrones de Material Design	Navegación sencilla y menús intuitivos	Gráficas útiles y con varios tipos de representación	Unidades más ingesta de alimentos	✓	✗	✗
Diario de diabetes mySgr	Interfaz cuidada, pero sin respetar patrones.	Muy rápida e intuitiva, destacando en vistazos rápidos la información	Gráficas de un único formato y poco integradas.	Opciones muy personalizables y con suficientes datos para añadir.	✓	✗	✓
Diabetes Monitor	Respeto los patrones de diseño, pero posee información demasiado detallada en cada vistazo, sobrecargando de datos al usuario.	La navegación es correcta, con menús diferenciados y representativos.	Posee representaciones gráficas intuitivas y explicativas.	Opciones personalizables y con datos representativos.	✓	✗	✓
Abbot LibreView	Simple y con menús explicativos para mejorar la facilidad de uso.	Debido a la sencillez de la aplicación, no requiere de demasiados menús, y por consiguiente la navegación es extremadamente sencilla.	Muy buena, posee gráficas muy explicativas con datos muy concretos.	Muy sencillo, pero necesita de un software de escritorio para relajar la carga de los datos desde un dispositivo propio de la marca.	✗	✓	✓

Tabla 4: Tabla resumen aplicaciones para el control de la glucemia

3-Análisis del sistema

La ingeniería de software incluye procesos, métodos y herramientas que permiten elaborar a tiempo y con calidad sistemas complejos basados en computadoras. El proceso de software incorpora cinco actividades estructurales: comunicación, planeación, modelado, construcción y despliegue que son aplicables a todos los proyectos de software. La práctica de la ingeniería de software es una actividad para resolver problemas, que sigue un conjunto de principios fundamentales. [24]

Esta es una de las múltiples definiciones que la Ingeniería de software ofrece a los desarrolladores y que acumula una serie de conceptos que nos serán de gran utilidad para el desarrollo de nuestro sistema. Como este, muchos antes han sido desarrollados siguiendo una cierta metodología, capaz de crear una sistema coherente, preciso, bien diseñado y sin errores, de tal forma que se reduzca en la mayor medida posible el impacto que pueda sufrir nuestro sistema durante todo su ciclo de vida.

De aquí en adelante, por tanto, vamos a tratar de definir cuál ha sido nuestra ingeniería del software, en todos y cada uno de los aspectos que esta cubre, haciendo que nuestro software este lo más adaptado posible a la misma, para justificar cada punto de su desarrollo.

3.1- Especificación de requisitos software

La especificación de requisitos software, también denominada ERS por sus siglas, podría definirse como el conjunto de especificaciones únicas, descriptivas y características de un software desarrollado en un ambiente concreto. Este conjunto de especificaciones, serán descriptivos de cada software desarrollado, pues su modificación obtendrá como resultado un software diferente. Serán características pues ayudarán a definir de manera clara y concisa cada uno de los aspectos que tendrá nuestro software, haciendo que este respete lo anteriormente dictado, y que sea coherente, preciso, este bien diseñado y sea verificable mediante pruebas específicas. Por último, nuestro conjunto de especificaciones deberá ser único pues mostrará sin lugar a dudas, todos los aspectos estructurales que tendrá nuestro sistema, estructurando la arquitectura, y siendo la base de desarrollo de la misma.

Como ya se ha comentado anteriormente, nuestra arquitectura va a seguir una metodología para conseguir justificar cada uno de los aspectos tratados sobre la misma. De esta manera, para la definición de los requisitos software, vamos a seguir el estándar IEEE830-1998 [25] , pero de manera simplificada y tomando solo algunos de los apartados definidos en el mismo, eliminado los apartados en los que nuestro sistema no aplica.

3.1.1- Especificación de Requisitos Software (IEEE830-1998)

3.1.1.1- Propósito y alcance

El propósito de la definición de requisitos siguiendo el estándar IEEE830, se centra en la creación de una base sólida sobre la cual consolidar nuestra arquitectura de software, haciendo que el sistema pueda ser coherente, consistente, completo, trazable y verificable a través de los mismos.

Con esto se pretende además definir la audiencia a la que va dirigido esta sección de documento, que se centra en los lectores que pretendan desarrollar un sistema de software con una arquitectura y diseño similar a nuestro sistema. De esta manera, el desarrollador que tenga tal propósito, conocerá cuáles han sido los puntos seguidos en el diseño de nuestra arquitectura, conociendo la misma con la máxima precisión posible, justificando además todos los factores que han llevado a la misma.

Como se ha comentado antes, el sistema a desarrollar, denominado [\[R1\]](#) “**GDMSCCT: GLUCOSE DATA MONITORING SYSTEM AND DIABETES CONTROL TIPS**”, desarrollará un software capaz de controlar y monitorizar de manera eficiente y útil, los valores de las glucemias proporcionadas por pacientes de Diabetes tipo 1, permitiendo un nivel de abstracción mayor al alcanzable a través de otros métodos de registro y monitorización, estando este optimizado para un dispositivo móvil con un sistema operativo concreto.

3.1.1.2-Definiciones, acrónimos y abreviaturas

Durante esta sección, tendremos como objetivo la definición de los acrónimos, términos y abreviaturas que nos servirán para mantener la legibilidad de toda la especificación de los requisitos. Esto se produce ya que, durante el desarrollo de este apartado, trataremos diferentes tipos de nomenclaturas propias, y debemos permitir al lector, mantener un nivel de comprensión óptimo para comprender sin lugar a la ambigüedad, este punto crítico de nuestro desarrollo teórico. Los diferentes conceptos estarán englobados en una tabla, cuyos campos estarán estructurados de la siguiente manera:

Número de referencia	Tipo	Definición
[1] GDMSCCT	Acrónimo	Este acrónimo hace referencia al nombre del sistema desarrollado. Este posee las siglas de GLUCOSE DATA MONITORING SYSTEM AND DIABETES CONTROL TIPS.

[2]RAM	Acrónimo	<p>“Random Acces Memory”</p> <p>se trata de un tipo de memoria hardware, utilizada por el sistema operativo para realizar con mayor fluidez las operaciones de lectura y escritura al desarrollar su funcionalidad, permitiendo intercambios más rápidos, pero temporales.</p>
[3] Feedback	Definición	<p>Concepto que hace referencia a una crítica, mostrando aspectos correctamente desarrollados y aspectos a mejorar tras una exposición teórica.</p>
[4]RecylcerView	Definición	<p>Concepto del lenguaje Android que hace referencia a una lista dinámica de datos personalizables por el usuario en diseño, siguiendo este la utilización de tarjetas para su disposición.</p>
[5]RadioButton	Definición	<p>Concepto del lenguaje Android que hace referencia a un conjunto de botones o casillas seleccionables, sobre las cuales pueden asignarse acciones.</p>
[6]Toast	Definición	<p>Mensaje temporal y aclaratorio proporcionado por la interfaz gráfica del lenguaje Android, que permite avisar de errores o notificar acciones.</p>

[7] CU-X	Acrónimo	Hace referencia a la notación con la que se tratarán los casos de uso definidos en el sistema. Se sustituirá el término X por el número correspondiente al caso de uso en orden ascendente según su definición.
[8] Man in the middle	Definición	Tipo de ataque informático generado por la inclusión de un sistema no autorizado entre dos sistemas que pretenden establecer una comunicación.

Tabla 5: Tabla de definiciones, acrónimos y abreviaturas

3.1.1.3- Perspectiva de producto

El sistema desarrollado, denominado **GDMSCCT: GLUCOSE DATA MONITORING SYSTEM AND DIABETES CONTROL TIPS** se desarrollará como un sistema independiente de cualquier otro, sin necesidad dependiente de ninguno otro, a excepción del sistema operativo sobre el que se desarrolle, que será Android y estará basado en Linux.

3.1.1.4- Funcionalidades del producto

Con respecto a las funcionalidades que debe desarrollar nuestro sistema, estas se centran principalmente en registrar y monitorear las medidas de glucemia que el usuario pretenda. De esta manera nuestro sistema tratará de aportar al usuario que lo utilice, un nivel de abstracción mayor sobre los niveles glucémicos que se almacenen, permitiendo al mismo mejorar sustancialmente el control de la enfermedad que padece, la Diabetes Tipo 1.

Este nivel de abstracción se conseguirá a través de un listado de medidas, ordenadas y representadas con diferentes formatos, así como la hora y la fecha de dicha medición, para permitir al usuario una mejor organización. En conjunción a este listado el sistema ofrecerá una serie de gráficas en diferentes formatos, que aportarán un conjunto de datos de interés para el control de las medidas, por parte del usuario. En ellas se recogerán los datos que el usuario ha ido introduciendo, y se distribuirán para crear dichos diseños, aportando información clave sobre la información. Por otro lado, y también como ayuda al usuario, se permitirá llevar a cabo un cálculo de varios valores como son el factor de sensibilidad y el ratio de insulina-carbohidratos, que son de especial interés para los pacientes que padezcan la enfermedad.

Por otro lado, la aplicación almacenará, en caso de que el usuario lo desee, el conjunto de citas médicas que tenga programadas dicho paciente con su médico. Esto

permitirá al mismo registrar los datos del hospital, doctor fecha y hora de la cita, para poder controlar a modo de agenda privada estos datos.

3.1.1.4- Características de los usuarios

Tipo de usuario	Cualquier usuario capaz de manejar un dispositivo móvil, con conocimiento del idioma del sistema.
Formación	Ninguna.
Habilidades	Sin necesidad.
Actividades	Capacidad de legibilidad para la adición de datos sobre los cuales pueda trabajar, en los campos específicos.

Tabla 6: Tabla de usuarios

Los usuarios hacia los que está enfocada este sistema software serán todos aquellos que padezcan la enfermedad de la Diabetes Tipo 1, con un dispositivo móvil a su disposición, y con la suficiente capacidad para elegir que datos han de disponerse en cada uno de los campos habilitados para ello.

La aplicación ha tratado de desarrollarse de tal manera que la curva de aprendizaje sea mínima para el usuario, siendo la simplicidad y sencillez de uso, un factor clave a la hora de utilizarla.

3.1.1.5- Restricciones

Durante el desarrollo de este apartado, se definirán algunas de las restricciones que nuestro sistema ha de tener, a la hora de realizar el desarrollo de su diseño.

Con respecto al sistema operativo, como se ha determinado anteriormente, este ha de ser un sistema Android, debido a que uno de los objetivos de nuestro sistema software, se centra en conseguir la mayor portabilidad y la aplicación a dispositivos móviles. Por otro lado, debido a las limitaciones hardware de nuestro dispositivo, deben poder mover con fluidez el sistema operativo y las transiciones entre páginas, por lo que se recomienda un mínimo hardware de un gigabyte de memoria RAM [\[R2\]](#) en el dispositivo. El sistema también debe contar con almacenamiento interno suficiente para guardar datos sobre los registros glucémicos, con un mínimo de doscientos megabytes de memoria interna libres.

Otro de los aspectos importantes se centra en que nuestro dispositivo, en el cual vayamos a utilizar el sistema, debe disponer de capacidades multitáctiles para poder favorecer los cambios entre menús de una manera sencilla e intuitiva. Esta se realizará mediante pulsaciones en diferentes iconos de la aplicación, así como permitiendo la introducción de campos de texto varios.

A la hora de especificar restricciones sobre un apartado más técnico de la aplicación, diremos que el lenguaje sobre el cual se desarrollará la misma será el lenguaje propio

de ANDROID, que esta a su vez basado en Java y XML, y el entorno de programación será el entorno de Android Studio, proporcionado por la empresa Google Inc.

Por último y denotando restricciones sobre el usuario, debemos decir que este ha de contar con cualquier tipo de dispositivo externo a la aplicación para poder medir y obtener las medidas de glucemia, presentes en su corriente sanguínea, en un determinado tiempo, para de esta manera ser capaces de después registrarlas en la aplicación y permitir que esta desarrolle toda su funcionalidad.

3.1.1.6-Suposiciones y dependencias

Para el diseño y desarrollo de nuestro sistema, hemos de desarrollar un conjunto de requisitos específicos de nuestro sistema. Este conjunto es la definición clara y concreta de las capacidades de un sistema software, por lo que existen aspectos críticos que, en caso de ser modificados, alterarían dicho conjunto de requisitos. En nuestro caso uno de esos factores es sin duda el sistema operativo en el que se vaya a utilizar nuestro software. Esto arrastraría una serie de modificaciones, muy evidentes a nivel de desarrollo, pues el lenguaje de programación en el que se debe implementar sería totalmente diferente.

Otro aspecto que haría variar en gran medida nuestro conjunto de requisitos es el público o los usuarios a los que va enfocada la aplicación. Esto influye ya que el conjunto al que se orienta nuestra aplicación es muy concreto, haciendo que tenga unos requerimientos concretos, y padezca una enfermedad muy particular, por lo que, al alterar dicho factor, el total de los requisitos deberían estar enfocados de una manera diferente.

3.1.1.7- Evolución previsible del sistema

El sistema software desarrollado posee sin ninguna duda una de las características generales de un desarrollo software tradicional, y es que el ciclo de vida de este tipo de sistemas no puede especificarse de manera demasiado concisa. Es aquí donde se introducirá un concepto referido a esta implementación iterativa: el concepto de “beta continua”. Este término hace referencia a la necesidad de retroalimentación continua que poseen los sistemas software, siendo el usuario final una pieza clave a la hora de aportar un feedback [\[R3\]](#) tras la utilización del mismo.

Este concepto permitirá a los desarrolladores el implementar nuevas funcionalidades demandadas, mejorar las ya creadas o eliminar las menos utilizadas. Este concepto se ve potenciado con la capacidad de interacción de los usuarios y los desarrolladores de aplicaciones, que, a través de foros, tiendas de aplicaciones (Google Play, Apple Store...) o contacto directo, permitirán este tipo de aportes para la mejora conjunta del sistema software.

Por otro lado, y dado que este sistema ha de potenciarse lo máximo posible, como se introdujo anteriormente, la intención es claramente la de llegar al mayor número de usuarios, permitiendo mediante la liberación del código fuente, el mejorar las funcionalidades por parte de otros desarrolladores, haciendo que se cree una

comunidad global de desarrolladores con la misma aplicación como fin, mejorando así, en definitiva, la calidad de nuestro software.

3.2- Requisitos específicos del sistema

Durante este apartado se van a tratar la lista de requisitos del sistema necesarios para el desarrollo del sistema especificado anteriormente. Esta lista será clara y concisa, permitiendo definir una arquitectura coherente y sólida [26].

El formato que seguirán los requisitos vendrá dado por el que sigue:

La siguiente tabla será la plantilla para las tablas de requisitos, en ella se especificarán un conjunto de campos que ayudarán a diferenciar sin ninguna duda los requisitos. Los campos contendrán el siguiente formato, que identificará:

- **Identificador de requisito:** En este campo se especificará, con una nomenclatura concreta y especificada en la tabla, la identificación del requisito.
- **Nombre del requisito:** Será una identificación textual del requisito. Se utiliza una breve descripción del requisito que permita identificarlo del resto.
- **Tipo:** Hace referencia al tipo de requisito que se está tratando. En caso de tratarse de una restricción, esta deberá estar definida a priori por el cliente.
- **La fuente del requisito:** Este campo mostrará el origen teórico del requisito especificado, es decir, la restricción o necesidad de que el requisito exista.
- **Prioridad:** En este campo se reflejará de mayor a menor la capacidad del requisito para afectar a nuestro sistema, siendo un valor con grado más alto, una influencia mayor en el sistema en caso de no desarrollarse.

Identificador de requisito	Requisito X
Nombre de requisito	
Tipo	<input checked="" type="checkbox"/> Requisito <input checked="" type="checkbox"/> Restricción
Fuente del requisito	
Prioridad del requisito	<input type="checkbox"/> Alto/Eencial <input checked="" type="checkbox"/> Medio/Deseado <input type="checkbox"/> Bajo/Opcional

Tabla 7: Plantilla de definición de requisitos

Tras esta definición resumida a modo de tabla, existirá una definición del requisito de manera descriptiva, donde se explicarán aspectos no reflejados antes y aclaraciones acerca de cada uno de los requisitos definidos.

Además de esto, para la correcta definición y organización de los requisitos, estos han sido categorizados entre los requisitos definidos como usuario de la aplicación, es decir necesidades propias del usuario que utilizará la aplicación. Tras esto existirá una distinción entre los requisitos funcionales y no funcionales, siendo los primeros

las posibilidades de realización en la aplicación, y la segunda los aspectos más cualitativos del sistema desarrollado.

3.2.1- Requisitos comunes a las interfaces

3.2.1.1-Interfaces de usuario

Identificador de requisito	Requisito 1
Nombre de requisito	Las interfaces de la aplicación deben ser descriptivas a su funcionalidad.
Tipo	<input type="checkbox"/> Requisito <input checked="" type="checkbox"/> Restricción
Fuente del requisito	Definición del cliente para el desarrollo del sistema.
Prioridad del requisito	<input checked="" type="checkbox"/> Alto/Esencial <input type="checkbox"/> Medio/Deseado <input type="checkbox"/> Bajo/Opcional

Tabla 8: Requisito 1

Este requisito funcional hace referencia al conjunto de iconos que se van a utilizar en nuestra interfaz del sistema. Para su definición es necesaria la utilización de iconos descriptivos, es decir, un icono que permita al usuario interpretar de una manera clara y concisa cual es la función que tiene dicho elemento visual en la interfaz. A modo de ejemplo, se puede aplicar un icono a los botones de navegación que permitan al usuario saber a qué pantalla del sistema dirigirá en caso de ser pulsado.

Identificador de requisito	Requisito 2
Nombre de requisito	La interfaz de usuario estará desarrollada para utilizarse de manera vertical.
Tipo	<input type="checkbox"/> Requisito <input checked="" type="checkbox"/> Restricción
Fuente del requisito	Definición del cliente para el desarrollo del sistema.
Prioridad del requisito	<input checked="" type="checkbox"/> Alto/Esencial <input type="checkbox"/> Medio/Deseado <input type="checkbox"/> Bajo/Opcional

Tabla 9: Requisito 2

Este requisito hacer referencia a la orientación de la pantalla del sistema. Esta está desarrollada de manera vertical, respetando la organización y posición de los elementos, ya que esta es representativa para cumplir su funcionalidad. Por tanto, esta debe estar restringida a su orientación vertical.

Identificador de requisito	Requisito 3
Nombre de requisito	Las interfaces de la aplicación deben ser amigables para su utilización aplicada a la tecnología táctil.
Tipo	<input type="checkbox"/> Requisito <input checked="" type="checkbox"/> Restricción
Fuente del requisito	Definición del cliente para el desarrollo del sistema.

Prioridad del requisito	<input checked="" type="checkbox"/> Alto/Eencial <input type="checkbox"/> Medio/Deseado <input type="checkbox"/> Bajo/Opcional
--------------------------------	--

Tabla 10: Requisito 3

3.2.1.2-Interfaces de hardware

Identificador de requisito	Requisito 4
Nombre de requisito	Las interfaces de la aplicación deben estar adaptadas a tamaños de pantalla de entre 5 pulgadas y 5,5 pulgadas.
Tipo	<input type="checkbox"/> Requisito <input checked="" type="checkbox"/> Restricción
Fuente del requisito	Definición del cliente para el desarrollo del sistema.
Prioridad del requisito	<input checked="" type="checkbox"/> Alto/Eencial <input type="checkbox"/> Medio/Deseado <input type="checkbox"/> Bajo/Opcional

Tabla 11: Requisito 4

Este requisito hace referencia a la creación de nuestra interfaz gráfica, y especifica claramente que esta no ha de estar optimizada en dispositivos mayores de 5,5 pulgadas de pantalla, ni menos de 5 pulgadas. Ya que el sistema está enfocado a ser lo más portable posible, utilizando para ellos su utilización en dispositivos móviles, se ha decidido desarrollarla para el conjunto de dispositivos móviles común en el mercado actual. Es por tanto que la interfaz no está diseñada para trabajar en formatos de tableta o dispositivos con tamaño reducido, ya que perderían aspectos visuales.

Identificador de requisito	Requisito 5
Nombre de requisito	La aplicación utilizará como memoria principal de almacenamiento la proporcionada por el dispositivo en el que está instalado.
Tipo	<input type="checkbox"/> Requisito <input checked="" type="checkbox"/> Restricción
Fuente del requisito	Definición del cliente para el desarrollo del sistema.
Prioridad del requisito	<input checked="" type="checkbox"/> Alto/Eencial <input type="checkbox"/> Medio/Deseado <input type="checkbox"/> Bajo/Opcional

Tabla 12: Requisito 5

Este requisito hace referencia a la necesidad de utilización de memoria de nuestro sistema. Este aprovechará la memoria interna libre del terminal para ser instalada, por lo que será necesario contar con almacenamiento suficiente para esto.

3.2.1.3-Interfaces de software

Identificador de requisito	Requisito 6
Nombre de requisito	La aplicación será desarrollada en el sistema operativo móvil Android.
Tipo	<input type="checkbox"/> Requisito <input checked="" type="checkbox"/> Restricción
Fuente del requisito	Definición del cliente para el desarrollo del sistema.
Prioridad del requisito	<input checked="" type="checkbox"/> Alto/Eencial <input type="checkbox"/> Medio/Deseado <input type="checkbox"/> Bajo/Opcional

Tabla 13: Requisito 6

Este requisito hace referencia al sistema operativo para el cual está desarrollada la aplicación. En este caso es el sistema operativo de software libre Android, y únicamente será posible su uso en dispositivos móviles que dispongan de tal sistema instalado.

3.2.1.3-Interfaces de comunicación

Identificador de requisito	Requisito 7
Nombre de requisito	La conexión entre los servidores para la realización de la copia de seguridad, estará cifrada de manera interna por el protocolo HTTPS.
Tipo	<input type="checkbox"/> Requisito <input checked="" type="checkbox"/> Restricción
Fuente del requisito	Definición del cliente para el desarrollo del sistema.
Prioridad del requisito	<input checked="" type="checkbox"/> Alto/Eencial <input type="checkbox"/> Medio/Deseado <input type="checkbox"/> Bajo/Opcional

Tabla 14: Requisito 7

Este requisito hace referencia al protocolo de conexión entre los servicios de la nube de Google Drive y nuestro sistema, que es el medio por el cual se almacenarán las copias de seguridad de los datos almacenados en la aplicación.

3.3- Requisitos no funcionales

3.3.1- Requisitos de rendimiento

Identificador de requisito	Requisito 8
Nombre de requisito	La muestra y visualización de los datos introducidos, obteniéndolos de la base de datos, han de mostrarse en un tiempo de entre uno y tres segundos.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción

Fuente del requisito	Necesario para un trabajo eficaz y seguro con la base de datos, así como para una experiencia de usuario satisfactoria, sin demasiados tiempos de espera.
Prioridad del requisito	<input type="checkbox"/> Alto/Eencial <input checked="" type="checkbox"/> Medio/Deseado <input type="checkbox"/> Bajo/Opcional

Tabla 15: Requisito 8

Este requisito hace referencia a los tiempos de carga que existirán en la aplicación a la hora de mostrar en la interfaz los datos almacenados en la base de datos de la misma. Esta ha de mostrar los registros en un tiempo marcado en el intervalo, evitando de esta manera que aplicaciones maliciosas puedan interferir en la conexión.

Identificador de requisito	Requisito 9
Nombre de requisito	La aplicación ha de mantener un nivel de consumo de memoria RAM óptimo para mantenerse corriendo en el dispositivo, evitando que el gestor de memoria del sistema cierre la aplicación por este hecho.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Necesario para mantener el rendimiento en el sistema y la no exhaustiva utilización de recursos del mismo.
Prioridad del requisito	<input type="checkbox"/> Alto/Eencial <input checked="" type="checkbox"/> Medio/Deseado <input type="checkbox"/> Bajo/Opcional

Tabla 16: Requisito 9

Este requisito hace referencia a las necesidades de nuestra aplicación a cumplir con un buen diseño, de tal manera que el consumo y gestión que esta haga de la memoria RAM del dispositivo, no sea un lastre en su utilización, consumiendo excesivos recursos del sistema y frustrando la experiencia de usuario.

3.3.2- Requisitos de seguridad

Identificador de requisito	Requisito 10
Nombre de requisito	El almacenamiento y lectura de los datos de la aplicación utilizarán el almacenamiento interno propio del dispositivo sin utilizar conexiones externas.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Necesario para garantizar la seguridad en las conexiones.
Prioridad del requisito	<input type="checkbox"/> Alto/Eencial <input checked="" type="checkbox"/> Medio/Deseado <input type="checkbox"/> Bajo/Opcional

Tabla 17: Requisito 10

Este requisito hace referencia a la utilización de conexiones externas a la aplicación para conectar con la base de datos. Esto se ha desarrollado de tal manera que no sea

necesaria dicho tipo de conexiones, evitando así posibles ataques de “*man in the middle*” en el tiempo de conexión. De esta manera, la base de datos será creada de manera automática por el sistema, permitiendo un acceso eficaz y seguro.

Identificador de requisito	Requisito 11
Nombre de requisito	La conexión con el sistema de copia de seguridad de Google, utilizará un túnel de cifrado HTTPS.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Necesario para garantizar la seguridad en las conexiones del sistema.
Prioridad del requisito	<input checked="" type="checkbox"/> Alto/Esencial <input type="checkbox"/> Medio/Deseado <input type="checkbox"/> Bajo/Opcional

Tabla 18: Requisito 11

Este requisito hace referencia a la utilización de un protocolo de seguridad para el cifrado de nuestras copias de seguridad. Aprovechando el entorno en el que se desarrolla el sistema, y la integración con los servicios propios de Google, cifraremos nuestros archivos almacenados en la nube mediante la política y sistema que actualmente utiliza la compañía, con cifrados HTTPS de alta seguridad, que garantizará conexiones seguras.

3.3.3- Requisitos de fiabilidad

Identificador de requisito	Requisito 12
Nombre de requisito	El sistema gestionará la aceptación de los permisos por parte del usuario, siendo este capaz de aceptarlo o declinarlo, en el momento en el que se requiera.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Necesario para mantener la privacidad del usuario.
Prioridad del requisito	<input type="checkbox"/> Alto/Esencial <input checked="" type="checkbox"/> Medio/Deseado <input type="checkbox"/> Bajo/Opcional

Tabla 19: Requisito 12

Este requisito hacer referencia al sistema de permisos que nuestra aplicación requerirá. En este caso, Android utiliza un sistema por defecto en el que se otorgan una serie de permisos de aplicación predeterminados, pero en caso de necesitar que el usuario otorgue requisitos específicos, estos deberán ser aceptados por el usuario.

3.3.4- Requisitos de disponibilidad

Identificador de requisito	Requisito 13
-----------------------------------	---------------------

Nombre de requisito	La aplicación contará con accesos a pantallas de información, para los datos que sean sensibles a la salud del paciente.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Necesario para garantizar la correcta utilización e información de la salud del paciente.
Prioridad del requisito	<input checked="" type="checkbox"/> Alto/Esencial <input type="checkbox"/> Medio/Deseado <input type="checkbox"/> Bajo/Opcional

Tabla 20: Requisito 13

Este requisito hace referencia a la disponibilidad que tendrá nuestro sistema para informar al usuario acerca de ciertos tipos de datos o informaciones mostradas que pueden ser sensibles a los valores de salud del usuario, pudiendo tener consecuencias directas en su salud.

Identificador de requisito	Requisito 14
Nombre de requisito	La aplicación mostrará tras su primer inicio, un tutorial de iniciación para el uso de la aplicación por parte del usuario final.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Necesario para una correcta utilización de la interfaz y las funcionalidades de nuestro sistema desarrollado.
Prioridad del requisito	<input type="checkbox"/> Alto/Esencial <input checked="" type="checkbox"/> Medio/Deseado <input type="checkbox"/> Bajo/Opcional

Tabla 21: Requisito 14

Este requisito hace referencia a la disponibilidad que existirá en nuestra aplicación de un tutorial en el momento en el que se inicie la aplicación por primera vez. En dicho tutorial se mostrarán los aspectos más relevantes de la aplicación, mostrando su utilización en los diferentes menús y las capacidades de la misma.

3.3.5- Requisitos de mantenibilidad

No se ha definido ningún requisito no funcional de este tipo.

3.3.6- Requisitos de portabilidad

Identificador de requisito	Requisito 15
Nombre de requisito	La aplicación deberá ser portable a cualquier dispositivo que cumpla con las especificaciones hardware requeridas, desempeñando además las mismas funcionalidades y con la misma eficacia.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción

Fuente del requisito	Necesario para mantener la portabilidad y rendimiento del sistema entre la variedad de dispositivos.
Prioridad del requisito	<input type="checkbox"/> Alto/Eencial <input checked="" type="checkbox"/> Medio/Deseado <input type="checkbox"/> Bajo/Opcional

Tabla 22:Requisito 15

Este requisito hacer referencia a la capacidad de la aplicación para funcionar de manera similar en diferentes dispositivos móviles, siempre que cumplan los requisitos específicos necesarios. De esta manera el sistema deberá funcionar de una manera similar en el total de estos dispositivos, con unos tiempos de carga y visualización similares en todas ellas, y similar disposición de su interfaz. Esto favorecerá la experiencia de usuario y la mayor disponibilidad para los usuarios.

3.4- Requisitos funcionales

3.4.1- Requisito funcional 1

Identificador de requisito	Requisito 16
Nombre de requisito	El sistema permitirá la creación y sobre escritura de un usuario en la base de datos.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	BD Usuario y menú Perfil.
Prioridad del requisito	<input checked="" type="checkbox"/> Alto/Eencial <input type="checkbox"/> Medio/Deseado <input type="checkbox"/> Bajo/Opcional

Tabla 23:Requisito 16

Este requisito hace referencia a la funcionalidad que tendrá nuestro sistema para identificar un conjunto de valores importante para identificar al usuario que la está utilizando, así como para obtener datos importantes para el cálculo de valores resultado.

Entradas

Las entradas para nuestra funcionalidad serán los campos de texto destinados al **nombre de usuario, apellidos de usuario, edad y el cálculo de unidades diarias de insulina utilizadas por el paciente**, situados en la pantalla de “Perfil”.

Procesos

Para desarrollar esta funcionalidad, el sistema debe establecer conexión con la base de datos, y concretamente realizar una inserción de los datos introducidos en cada uno de los campos. Esta inserción se realizará de tal manera que cada campo del cuadro de texto, corresponderá con su columna en la tabla **Usuarios**, que dispondrá de: **Nombre, Apellidos, Edad y UnidadesInsulina**. Esta acción se desarrollará en el momento en el que se realice una pulsación sobre el botón de guardar, definido en la interfaz.

Además de esto en caso de querer sobrescribir un usuario con un conjunto de datos nuevo, únicamente debemos introducir los datos de nuevo, para la evitación de errores, y se sobrescribirá en nuestra tabla con el nuevo conjunto.

Salidas

La comprobación de que no se han producido errores en el proceso, conlleva a que la aplicación devuelva al usuario a la pantalla principal de navegación.

En caso de error por falta de completitud en los campos de texto, ya que el sistema no aceptará valores nulos para este conjunto, la aplicación avisará al usuario de que existe dicho error mediante un mensaje tipo **Toast** [6], que se mostrará en la aplicación durante unos segundos.

En caso de producirse algún error en la conexión a la base de datos, el sistema de igual manera que en el caso anterior mostrará un mensaje tipo **Toast**, avisando sobre el problema al usuario para que trate de intentarlo de nuevo.

3.4.2- Requisito funcional 2

Identificador de requisito	Requisito 17
Nombre de requisito	El sistema permitirá la creación e inserción de citas clínicas, personalizadas por el usuario.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	BD Citas y menú Perfil.
Prioridad del requisito	<input checked="" type="checkbox"/> Alto/Esencial <input type="checkbox"/> Medio/Deseado <input type="checkbox"/> Bajo/Opcional

Tabla 24:Requisito 17

Este requisito hace referencia a la funcionalidad que tendrá nuestro sistema para crear citas personalizadas. Esto permitirá al usuario realizar una inserción de varios campos de texto, correspondientes con datos sobre dicha cita, almacenando estos después en una lista desplazable para mejorar su visualización.

Entradas

Las entradas para nuestra funcionalidad serán los campos de texto destinados al **nombre dl doctor, hospital de la consulta, fecha y hora de la cita y un campo dedicado para las notas sobre dicha cita**, haciéndose presente la visualización en forma de lista en la pantalla “Perfil”.

Procesos

Para desarrollar esta funcionalidad, el sistema debe establecer conexión con la base de datos, y concretamente realizar una inserción de los datos introducidos en cada uno de los campos. Esta inserción se realizará de tal manera que cada campo del cuadro de texto, corresponderá con su columna en la tabla **Citas**, que dispondrá de:

NombreDoctor, Hospital, Fecha, Hora y DescripcionCita. Esta acción de inserción se desarrollará en el momento en el que se realice una pulsación sobre el botón de guardar, definido en la interfaz.

Tras ello utilizaremos los datos almacenados para realizar la visualización de nuestra lista completa de citas registradas.

Salidas

La comprobación de que no se han producido errores en el proceso, conlleva a que la aplicación devuelva al usuario a la pantalla principal de navegación.

En caso de error por falta de completitud en los campos de texto, ya que el sistema no aceptará valores nulos para este conjunto, la aplicación avisará al usuario de que existe dicho error mediante un mensaje tipo **Toast**, que se mostrará en la aplicación durante unos segundos.

En caso de producirse algún error en la conexión a la base de datos, el sistema de igual manera que en el caso anterior mostrará un mensaje tipo **Toast**, avisando sobre el problema al usuario para que trate de intentarlo de nuevo.

Otra salida correspondiente a este conjunto de procesos es la visualización en forma de lista desplazable, también denominada en términos técnicos del lenguaje de programación **RecycledView**, en la pantalla de perfil.

3.4.3- Requisito funcional 3

Identificador de requisito	Requisito 18
Nombre de requisito	El sistema permitirá la creación e inserción de registros glucémicos.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	BD Medida y menú Registros.
Prioridad del requisito	<input checked="" type="checkbox"/> Alto/Esencial <input type="checkbox"/> Medio/Deseado <input type="checkbox"/> Bajo/Opcional

Tabla 25:Requisito 18

Este requisito hace referencia a la funcionalidad que tendrá nuestro sistema para crear registros de glucemia por parte del usuario. Esto permitirá al usuario realizar una inserción de varios campos de texto en una pantalla dedicada a ello, correspondientes con datos sobre dicho registro, con una serie de valores, almacenando estos después en una lista desplazable para mejorar su visualización.

Entradas

Las entradas para nuestra funcionalidad serán los campos de texto destinados a **medida de glucemia, fecha y hora de la medición y un campo dedicado para las**

notas sobre dicha cita, haciéndose presente la visualización en forma de lista en la pantalla **“Registros”**.

Procesos

Para desarrollar esta funcionalidad, el sistema debe establecer conexión con la base de datos, y concretamente realizar una inserción de los datos introducidos en cada uno de los campos. Esta inserción se realizará de tal manera que cada campo del cuadro de texto, corresponderá con su columna en la tabla **Medidas**, que dispondrá de: **Glucemia, Fecha, Hora y DescripcionMedida**. Esta acción de inserción se desarrollará en el momento en el que se realice una pulsación sobre el botón de guardar, definido en la interfaz.

Tras ello utilizaremos los datos almacenados para realizar la visualización de nuestra lista completa de registros glucémicos almacenados en la base de datos.

Salidas

La comprobación de que no se han producido errores en el proceso, conlleva a que la aplicación devuelva al usuario a la pantalla principal de navegación.

En caso de error por falta de completitud en los campos de texto, ya que el sistema no aceptará valores nulos para este conjunto, la aplicación avisará al usuario de que existe dicho error mediante un mensaje tipo **Toast**, que se mostrará en la aplicación durante unos segundos.

En caso de producirse algún error en la conexión a la base de datos, el sistema de igual manera que en el caso anterior mostrará un mensaje tipo **Toast**, avisando sobre el problema al usuario para que trate de intentarlo de nuevo.

Otra salida correspondiente a este conjunto de procesos es la visualización en forma de lista desplazable, también denominada en términos técnicos del lenguaje de programación **RecyclerView**, en la pantalla de registros.

3.4.4- Requisito funcional 4

Identificador de requisito	Requisito 19
Nombre de requisito	El sistema mostrará las citas y registros glucémicos en forma de lista deslizable.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	BD Medidas, BDCitas, menú registros, menú Perfil y RecyclerView [4] asociados a las listas de citas y registros glucémicos.
Prioridad del requisito	<input type="checkbox"/> Alto/Esencial <input checked="" type="checkbox"/> Medio/Deseado <input type="checkbox"/> Bajo/Opcional

Tabla 26:Requisito 19

Este requisito hace referencia a la funcionalidad que tendrá nuestro sistema para mostrar la visualización de los registros de glucemia registrados por parte del usuario. Esto permitirá al usuario obtener más información de manera más clara sobre los datos que ha introducido durante el uso de la aplicación.

Entradas

Las entradas para nuestra funcionalidad serán las tupas de las tablas Citas y de la tabla Medidas, almacenadas en nuestra base de datos.

Procesos

Para desarrollar esta funcionalidad, el sistema debe establecer conexión con la base de datos, y concretamente realizar una consulta de los datos almacenados en las tuplas de la tabla Medidas o de la tabla Citas según corresponda. En el momento en el que se obtenga uno, se mostrará en la lista desplazable dispuesta en el espacio de la interfaz destinada para esto, modificando aspectos gráficos de la interfaz como el color, los efectos de desplazamiento y los datos, dependiendo de los valores obtenidos. Esta acción de inserción se desarrollará de manera automática en el momento en el que se inicialice la pantalla de Registros o Perfil, y se obtengan los datos.

Salidas

En caso de producirse algún error en la conexión a la base de datos, el sistema mostrará un mensaje tipo **Toast**, avisando sobre el problema al usuario para que trate de intentarlo de nuevo.

La salida correspondiente a este conjunto de procesos es la visualización en forma de lista desplazable, también denominada en términos técnicos del lenguaje de programación **RecyclerView**, en la pantalla correspondiente.

3.4.5- Requisito funcional 5

Identificador de requisito	Requisito 20
Nombre de requisito	El sistema permitirá la consulta de los detalles de la descripción de las citas y registros glucémicos, realizando una pulsación sobre la tarjeta que lo contiene en la lista de visualización.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	BD Medidas, menú registros y RecyclerView asociado a la lista de registros glucémicos.

Prioridad del requisito	<input checked="" type="checkbox"/> Alto/Esencial <input type="checkbox"/> Medio/Deseado <input type="checkbox"/> Bajo/Opcional
--------------------------------	--

Tabla 27:Requisito 20

Este requisito hace referencia a la funcionalidad que tendrá nuestro sistema para mostrar la descripción o detalles almacenados en la base de datos, de manera dinámica y simplificada. Esto permitirá al usuario obtener más información sobre los datos visualizados en las listas de las pantallas de Perfil y de Registros.

Entradas

Las entradas para nuestra funcionalidad serán la pulsación realizada sobre las tarjetas que contendrán los datos en nuestras listas desplazables.

Procesos

Para desarrollar esta funcionalidad, el sistema debe crear un Listener capaz de detectar cuando se ha realizado una pulsación sobre un elemento de una lista o RecyclerView, creando en dicho momento una pantalla de información en la que se mostrarán los detalles introducidos por el usuario al insertar en la base de datos una cita o un registro glucémico. De esta manera se le permitirá obtener detalles más concretos o anotaciones sobre dicha inserción de una manera rápida y concreta.

Salidas

La salida correspondiente a este conjunto de procesos es la visualización, en forma de pantalla de información, de los datos que se hayan recopilado para esa cita, o registro, de la base de datos.

3.4.6- Requisito funcional 6

Identificador de requisito	Requisito 21
Nombre de requisito	El sistema permitirá la eliminación de las citas y registros glucémicos, realizando una pulsación larga sobre la tarjeta que lo contiene en la lista de visualización.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	BD Medidas, menú registros y RecyclerView asociado a la lista de registros glucémicos.
Prioridad del requisito	<input checked="" type="checkbox"/> Alto/Esencial <input type="checkbox"/> Medio/Deseado <input type="checkbox"/> Bajo/Opcional

Tabla 28:Requisito 21

Este requisito hace referencia a la funcionalidad que tendrá nuestro sistema para eliminar registros de nuestras listas, de manera dinámica y simplificada. Esto permitirá al usuario borrar información innecesaria sobre los datos visualizados en las listas de las pantallas de Perfil y de Registros.

Entradas

Las entradas para nuestra funcionalidad serán la pulsación prolongada, realizada sobre las tarjetas que contendrán los datos en nuestras listas desplazables. Tras la misma se deberá aceptar el mensaje de confirmación para proceder con el borrado del registro o cita.

Procesos

Para desarrollar esta funcionalidad, el sistema debe crear un Listener capaz de detectar cuando se ha realizado una pulsación larga sobre un elemento de una lista o RecyclerView, creando en dicho momento una pantalla de confirmación en la que se podrá aceptar o declinar el eliminar la cita o un registro glucémico. De esta manera se le permitirá borrar elementos de la lista, y por consiguiente de la base de datos, de una manera rápida y simplificada.

Salidas

La salida correspondiente a este conjunto de procesos es el retorno por parte del sistema a la página principal del sistema, junto con el mensaje de confirmación de la eliminación del registro o cita.

3.4.7- Requisito funcional 7

Identificador de requisito	Requisito 22
Nombre de requisito	El sistema mostrará gráficas de tipo pastel y gráficas lineales con los datos introducidos por el usuario como registros glucémicos.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	BD Medidas, menú Gráficas.
Prioridad del requisito	<input checked="" type="checkbox"/> Alto/Esencial <input type="checkbox"/> Medio/Deseado <input type="checkbox"/> Bajo/Opcional

Tabla 29:Requisito 22

Este requisito hace referencia a la funcionalidad que tendrá nuestro sistema para mostrar un conjunto de datos en forma de gráficas, siendo estas de varios tipos como son las de tipo pastel, y las gráficas lineales, combinando la distribución de valores glucémicos, o las medidas y tiempo, de manera correspondiente.

Entradas

Las entradas para nuestra funcionalidad serán los datos obtenidos de la base de datos y dispuestos en forma de lista en el menú de Registros.

Procesos

Para desarrollar esta funcionalidad, el sistema debe crear varias gráficas combinando los datos que se obtienen de la base de datos y de la lista de registros. De esta manera, utilizaremos una librería externa que nos permitirá, convirtiendo esos datos a los tipos necesarios, crear los diferentes tipos de gráficas.

Salidas

La salida correspondiente a este conjunto de procesos es el conjunto de gráficas especificadas, dispuestas en el menú de gráficas y que podrán ser visualizadas dependiendo de la selección de un RadioButton [5] específico para cambiar entre ambos tipos.

3.4.8- Requisito funcional 8

Identificador de requisito	Requisito 23
Nombre de requisito	El sistema permitirá cambiar y visualizar el tipo de graficas de manera dinámica.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Menú Gráficas y RadioButton.
Prioridad del requisito	<input checked="" type="checkbox"/> Alto/Esencial <input type="checkbox"/> Medio/Deseado <input type="checkbox"/> Bajo/Opcional

Tabla 30:Requisito 23

Este requisito hace referencia a la funcionalidad que tendrá nuestro sistema para mostrar un conjunto de datos en forma de gráficas, siendo estas de varios tipos como son las de tipo pastel, y las gráficas lineales, combinando la distribución de valores glucémicos, o las medidas y tiempo, de manera correspondiente al botón marcado en el RadioButton de la interfaz.

Entradas

La entrada de esta funcionalidad será el marcado de uno de los campos del RadioButton definido en la interfaz.

Procesos

Para desarrollar esta funcionalidad, el sistema debe crear varias gráficas combinando los datos que se obtienen de la base de datos y de la lista de registros. Con forme a la pulsación que realice el usuario sobre uno u otro tipo de graficas definidas en los RadioButton de la interfaz, se visualizará un tipo u otro.

Salidas

La salida correspondiente a este conjunto de procesos es el un tipo concreto de gráficas, conforme al botón que se haya marcado

3.4.9- Requisito funcional 9

Identificador de requisito	Requisito 24
Nombre de requisito	El sistema realizará una copia de seguridad automática en Google Drive todos los días a una hora concreta
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Archivos de datos almacenados en la memoria local del dispositivo
Prioridad del requisito	<input checked="" type="checkbox"/> Alto/Esencial <input type="checkbox"/> Medio/Deseado <input type="checkbox"/> Bajo/Opcional

Tabla 31:Requisito 24

Este requisito hace referencia a la funcionalidad que tendrá nuestro sistema para realizar una copia de seguridad de manera automática en el servicio de almacenamiento en la nube de Google Drive. Estas copias se realizarán de manera automática todos los días a las 2:00 horas de la noche.

Entradas

La entrada de esta funcionalidad será el archivo de sistema que contendría los datos almacenados por el usuario durante el día.

Procesos

Para desarrollar esta funcionalidad, el sistema debe mantener una cuenta de Google, que se activará de manera automática gracias a la cuenta con la que se ha iniciado sesión en el dispositivo. De esta manera cada día será capaz de manera automática de realizar una copia de seguridad, subiendo el archivo al servicio cloud ofrecido por Google, y que permitirá al usuario restablecer datos en caso de pérdida o necesidad.

Salidas

La salida correspondiente a este conjunto de procesos es la creación de manera automática de los datos almacenados en el dispositivo por parte del usuario.

3.4.10- Requisito funcional 10

Identificador de requisito	Requisito 25
Nombre de requisito	El sistema restaurará los datos almacenados en la copia de seguridad cuando se pulse el botón definido para esto
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Botón de recuperación de copia de seguridad

Prioridad del requisito	<input type="checkbox"/> Alto/Eencial	<input checked="" type="checkbox"/> Medio/Deseado	<input type="checkbox"/> Bajo/Opcional
--------------------------------	---------------------------------------	---	--

Tabla 32: Requisito 25

Este requisito hace referencia a la funcionalidad que tendrá nuestro sistema para restaurar los datos almacenados en el servicio de almacenamiento en la nube de Google Drive. Estas copias se realizarán cuando el usuario utilitario lo precise.

Entradas

La entrada de esta funcionalidad será el archivo de copia de seguridad almacenado en la nube

Procesos

Para desarrollar esta funcionalidad, el sistema debe mantener una cuenta de Google, que se activará de manera automática gracias a la cuenta con la que se ha iniciado sesión en el dispositivo. De esta manera cuando el usuario desee recuperar sus datos, ya sea por perdida o necesidad, pulsará el botón destinado para ello en la interfaz.

Salidas

La salida correspondiente a este conjunto de procesos es la restauración de los datos respaldados en la nube de manera automática por el sistema.

3.5- Otros requisitos

En este apartado se definirán los requisitos que no pueden ser categorizados en ninguno de los demás grupos. La notación que seguiremos para ellos será exactamente la misma que para los requisitos no funcionales mostrados anteriormente.

3.5.1- Requisitos culturales y políticos

No se ha definido ningún requisito de este tipo.

3.5.2- Requisitos legales

Identificador de requisito	Requisito 26
Nombre de requisito	El acuerdo de términos y condiciones del sistema especificará los datos utilizados y guardados por la aplicación, y se hará saber del mismo al usuario que utilice la aplicación.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Necesario para mantener la privacidad del usuario utilitario del sistema.
Prioridad del requisito	<input checked="" type="checkbox"/> Alto/Eencial <input type="checkbox"/> Medio/Deseado <input type="checkbox"/> Bajo/Opcional

Tabla 33: Requisito 26

Este requisito hace referencia a la creación de un documento que reflejará los términos y condiciones, especificando en ellos los datos que van a ser utilizados y el uso de los mismos, bajo la responsabilidad del usuario.

3.6-Casos de uso

Pero durante el desarrollo de nuestro sistema, existen un conjunto de descripciones que nos permitirá saber como interactúa el sistema con nuestro usuario final. De esta manera, se describen casos particulares de utilización, de tal manera que se pueda aplicar una nueva perspectiva de funcionamiento para llevar a cabo la implementación a posteriori de nuestro sistema [27].

Nuestro sistema concretamente poseerá un conjunto definido de casos de usos, entre los que se pueden destacar funcionalidades propias de nuestro sistema, tales como la **inserción de citas o registros glucémicos**, la **eliminación de citas o registros glucémicos** y la **visualización de mensajes importantes por el sistema**.

Este conjunto de casos de uso nos permitirá además abstraernos del nivel real de desarrollo de la aplicación, exponiendo una vista del sistema simplificada, y de qué debe hacer este durante su funcionamiento.

Como hemos desarrollado en anteriores puntos de este documento, trataremos de seguir una notación o desarrollo particular para especificar cada uno de los casos de uso que hemos definido para nuestro sistema. Así pues, los casos de uso serán especificados en forma de tabla en primera instancia, seguidos de un breve resumen y el diagrama que corresponde al mismo después. Por otro lado, debemos añadir que el lenguaje que utilizaremos para la definición de los diagramas será el lenguaje UML [28] [29].

La tabla que completaremos con cada caso de uso seguirá el siguiente formato:

CU-X [7]	Nombre del caso de uso
Objetivos	Objetivos del caso de uso
Requisitos asociados	Requisitos que influyen la definición del caso de uso
Descripción	
Precondición	Necesidades que deben cumplirse para que se lleve a cabo dicha acción.

Secuencia Normal	Paso	Acción
	X	
Postcondición	Necesidades que deben cumplirse para que se lleve a cabo dicha acción.	
Excepciones	Paso	Acción
	X	
Rendimiento	Paso	Cota de tiempo
	X	
Frecuencia esperada	Frecuencia con la que se espera que se desarrolle la acción. Debe ser un valor cuantitativo.	
Comentarios	Comentarios adicionales	

Tabla 34: Plantilla de tabla de casos de uso

3.6.1- Caso de uso 1: Creación de un usuario en el sistema

CU-01	Creación de usuario
Objetivos	El objetivo principal de este caso de uso es el de registrar un usuario en el sistema para que pueda utilizar la funcionalidad total de la aplicación
Requisitos asociados	Requisito 16
Descripción	Este caso mostrará como es el proceso para la creación de un nuevo usuario en el sistema, haciendo que el mismo quede registrado en la base de datos para comenzar a recopilar sus datos clínicos, y que después estos puedan ser monitoreados.

Precondición	La única necesidad que se posee para este caso de uso es la de haber iniciado la aplicación, sin que los datos almacenados en la base de datos contengan ningún usuario anteriormente creado	
Secuencia Normal	Paso	Acción
	1	El usuario inicia desde el sistema operativo la aplicación por primera vez.
	2	El sistema conducirá al usuario a través del tutorial, mostrando los diferentes menús de la que esta dispone.
	3	El usuario, tras terminar el tutorial, pulsará sobre el botón identificado por el icono de Perfil
	4	El sistema reconducirá al usuario a la nueva pantalla de perfil, donde existirán un conjunto de campos de texto que este debe rellenar
	5	El usuario completará los datos que se le piden: Nombre, Apellidos, Edad y Unidades de glucemia diarias.
	6	El usuario pulsara sobre el botón de guardar, mostrado en la interfaz.
	7	El sistema devolverá al usuario a la pantalla principal para que pueda seguir utilizando la aplicación.
Postcondición	El usuario pasa a estar registrado en la base de datos y podrá utilizar todas las funciones del sistema sin limitaciones.	
Excepciones	Paso	Acción
	5	E.1- Si el usuario no completa o lo hace de manera incorrecta alguno de los campos dispuestos, el sistema notificará al mismo sobre dicho error mediante un mensaje Toast.
	6	E.2- En caso de que se produzca algún error al conectar con la base de datos, esta devolverá un mensaje de aviso al usuario para que realice un nuevo intento pasado un tiempo
Rendimiento	Paso	Cota de tiempo
	6	1-3 segundos

Frecuencia esperada	1
Comentarios	Sin comentarios adicionales

Tabla 35: Caso de uso 1

3.6.2-Caso de uso 2: Inserción de una cita médica

CU-02	Inserción de una cita médica	
Objetivos	El objetivo principal de este caso de uso es el de permitir al usuario insertar una cita, concretada anteriormente con su médico, que funcione a modo de recordatorio en su perfil	
Requisitos asociados	Requisito 17- Requisito 19	
Descripción	Este caso de uso mostrará como es el proceso para la creación de una cita médica personalizada por parte del usuario, y su posterior adición a la lista desplazable destinada para su visualización.	
Precondición	La única necesidad que se posee para este caso de uso es la de haber iniciado la aplicación y que se haya creado con anterioridad un usuario utilitario de la misma	
Secuencia Normal	Paso	Acción
	1	El usuario inicia desde el sistema operativo la aplicación
	2	El sistema mostrará la pantalla principal de navegación al usuario tras su inicio
	3	El usuario pulsará sobre el botón identificado por el icono de Perfil
	4	El sistema reconducirá al usuario a la nueva pantalla de Perfil, donde existirán un botón flotante identificado con un símbolo "+" donde este podrá añadir una nueva cita
	5	El sistema navegará a una nueva pantalla en la que se dispondrán diferentes campos de texto y de selección para que el usuario concrete su cita de la manera más precisa posible

	6	El usuario completará con los datos personalizados, cada uno de los campos definidos: Nombre del Doctor, Hospital donde se realizará la consulta, Fecha, Hora y Descripción o notas sobre la consulta , y pulsará el botón de guardar dispuesto en la interfaz.
	7	El sistema devolverá al usuario a la pantalla principal para que pueda seguir utilizando la aplicación, con el mensaje de verificación que hará saber que la cita ha sido añadida .
Postcondición	Los datos de la cita pasan a estar registrados en la base de datos, y podrán visualizarse en la lista de visualización específica para ello.	
Excepciones	Paso	Acción
	6	E.1- Si el usuario no completa o lo hace de manera incorrecta alguno de los campos dispuestos para concretar la cita, el sistema notificará al mismo sobre dicho error mediante un mensaje Toast.
	7	E.2- En caso de que se produzca algún error al conectar con la base de datos, esta devolverá un mensaje de aviso al usuario para que realice un nuevo intento pasado un tiempo
Rendimiento	Paso	Cota de tiempo
	6	1-3 segundos
Frecuencia esperada	1 cita cada 3 meses	
Comentarios	La frecuencia con la que una cita es concretada con el médico depende mucho de la edad de los pacientes, así como del tratamiento que este siga. Por norma general 3 meses es un tiempo suficiente para pulir imperfecciones reseñables en la salud del paciente, pero este es un dato muy personal y variable dependiendo del tipo de paciente y sus características.	

Tabla 36: Caso de uso 2

3.6.3-Caso de uso 3: Registrar una medida glucémica

CU-03	Insertión de un registro glucémico
Objetivos	El objetivo principal de este caso de uso es el de permitir al usuario insertar un registro glucémico concreto, tras haber obtenido dicho valor en la medición de su glucemia de manera externa

Requisitos asociados	Requisito 18- Requisito 19	
Descripción	Este caso de uso mostrará como es el proceso para la creación en el sistema de una medida glucémica personalizada por parte del usuario, y su posterior adición a la lista desplazable destinada para su visualización.	
Precondición	Las necesidades que se deben cumplir para este caso de uso es la de haber iniciado la aplicación, que se haya creado con anterioridad un usuario utilitario de la misma y haber realizado la medición de la glucemia del paciente con algún sistema de medición externo	
Secuencia Normal	Paso	Acción
	1	El usuario inicia desde el sistema operativo la aplicación
	2	El sistema mostrará la pantalla principal de navegación al usuario tras su inicio
	3	El usuario pulsará sobre el botón identificado por el icono de Registros
	4	El sistema reconducirá al usuario a la nueva pantalla de Registros, donde existirán un botón flotante identificado con un símbolo "+" donde este podrá añadir una nueva medida glucémica
	5	El sistema navegará a una nueva pantalla en la que se dispondrán diferentes campos de texto y de selección para que el usuario concrete su glucemia de la manera más precisa posible
	6	El usuario completará con los datos personalizados, cada uno de los campos definidos: Glucemia, Fecha, Hora, Descripción o notas sobre la medida y pulsará el botón de guardar dispuesto en la interfaz.
	7	El sistema devolverá al usuario a la pantalla principal para que pueda seguir utilizando la aplicación, con el mensaje de verificación que hará saber que el registro ha sido añadida.
Postcondición	Los datos de la medida pasan a estar registrados en la base de datos, y podrán visualizarse en la lista de visualización específica para ello.	
Excepciones	Paso	Acción

	6	E.1- Si el usuario no completa o lo hace de manera incorrecta, alguno de los campos dispuestos para concretar la cita, el sistema notificará al mismo sobre dicho error mediante un mensaje Toast.
	7	E.2- En caso de que se produzca algún error al conectar con la base de datos, esta devolverá un mensaje de aviso al usuario para que realice un nuevo intento pasado un tiempo
Rendimiento	Paso	Cota de tiempo
	6	1-3 segundos
Frecuencia esperada	1 medida cada 4 horas, es decir aproximadamente 6 medidas al día	
Comentarios	Esta frecuencia de medición es muy relativa, y variará en gran medida dependiendo de las necesidades del paciente y sus tendencias glucémicas. Es difícil controlar este aspecto puesto que las competiciones deportivas, periodos de estrés alto, o situaciones poco comunes pueden requerir de una frecuencia mayor de medición, por lo que se han precisado las mediciones mínimas recomendadas.	

Tabla 37: Caso de uso 3

3.6.4-Caso de uso 4: Mostrar detalles de una lista deslizable

CU-04	Mostrar detalles almacenados para un registro, desde su lista deslizable correspondiente	
Objetivos	El objetivo principal de este caso de uso es el de permitir al usuario conocer con más detalle los valores de las descripción o notas almacenados tanto en las citas, como en los registros de glucemia en el momento de su adición en el sistema	
Requisitos asociados	Requisito 17-Requisito 18-Requisito 19-Requisito 20	
Descripción	Este caso de uso mostrará como es el proceso para visualizar los datos de una lista, realizando una pulsación sobre la tarjeta que contiene la medida o la cita requerida	
Precondición	Las necesidades que se deben cumplir para este caso de uso es la de haber iniciado la aplicación, que se haya creado con anterioridad un usuario utilitario de la misma y haber realizado la inserción de una cita o el registro glucémico sobre la que se quiera realizar la consulta	
Secuencia	Paso	Acción

Normal	1	El usuario inicia desde el sistema operativo la aplicación
	2	El sistema mostrará la pantalla principal de navegación al usuario tras su inicio
	3	El usuario pulsará sobre el botón identificado por el icono de Registros//Perfil dependiendo de qué detalles se quieran conocer, siendo sobre una medición o una cita respectivamente
	4	El sistema reconducirá al usuario a la nueva pantalla de Registros//Perfil, donde se dispondrá la lista deslizable con los valores anteriormente añadidos
	5	El usuario realizará una pulsación precisando la tarjeta de la lista sobre la que se desea conocer los detalles
	6	El sistema mostrará una nueva pantalla sobre la interfaz con los detalles almacenados para dicha cita o medida
	7	El usuario podrá pulsar cualquier parte de la nueva pantalla para retornar al punto anterior y navegar con normalidad
Postcondición	Los datos almacenados en la medida o registro pasan a mostrarse en la pantalla durante el tiempo que el usuario considere	
Excepciones	Paso	Acción
	6	E.1- En caso de que se produzca algún error al conectar con la base de datos, esta devolverá un mensaje de aviso al usuario para que realice un nuevo intento pasado un tiempo
Rendimiento	Paso	Cota de tiempo
	6	Determinado por el usuario
Frecuencia esperada	Sin frecuencia previsible	
Comentarios	Sin comentario adicionales	

Tabla 38: Caso de uso 4

3.6.5-Caso de uso 5: Eliminar registros y citas del sistema

CU-05	Eliminar los datos de las citas o registros almacenados anteriormente en el sistema	
Objetivos	El objetivo principal de este caso de uso es el de permitir al usuario eliminar del sistema los registros y citas médicas que anteriormente se hayan insertado, en caso de error o defecto de los mismos	
Requisitos asociados	Requisito 17-Requisito 18-Requisito 19-Requisito 21	
Descripción	Este caso de uso mostrará como es el proceso para eliminar los datos del sistema desde la lista deslizable concreta, realizando una pulsación mantenida sobre la tarjeta que contiene la medida o la cita que se desea eliminar	
Precondición	Las necesidades que se deben cumplir para este caso de uso es la de haber iniciado la aplicación, que se haya creado con anterioridad un usuario utilitario de la misma y haber realizado la inserción de una cita o el registro glucémico sobre la que se quiera realizar la eliminación	
Secuencia Normal	Paso	Acción
	1	El usuario inicia desde el sistema operativo la aplicación
	2	El sistema mostrará la pantalla principal de navegación al usuario tras su inicio
	3	El usuario pulsará sobre el botón identificado por el icono de Registros//Perfil dependiendo de qué detalles se quieran conocer, siendo sobre una medición o una cita respectivamente
	4	El sistema reconducirá al usuario a la nueva pantalla de Registros//Perfil, donde se dispondrá la lista deslizable con los valores anteriormente añadidos
	5	El usuario realizará una pulsación mantenida precisando la tarjeta de la lista sobre la que se desea eliminar los datos
	6	El sistema mostrará una nueva pantalla sobre la interfaz con la confirmación de eliminación de la cita o medida

	7	El usuario pulsara el botón de confirmación para eliminar la media, o el de rechazo para volver a la pantalla anterior
	8	El sistema retornará al usuario a la pantalla principal en caso de confirmar la eliminación
Postcondición	Los datos almacenados en la medida o registro pasan eliminarse del sistema, eliminando sus valores en la base de datos	
Excepciones	Paso	Acción
	7	E.1- En caso de que se produzca algún error al conectar con la base de datos, esta devolverá un mensaje de aviso al usuario para que realice un nuevo intento pasado un tiempo
Rendimiento	Paso	Cota de tiempo
	7	1-3 segundos
Frecuencia esperada	Sin frecuencia previsible	
Comentarios	Sin comentario adicionales	

Tabla 39: Caso de uso 5

3.6.6-Caso de uso 6: Mostrar gráficas de diferentes tipos

CU-06	Mostrar las diferentes gráficas para los datos registrados en nuestro sistema
Objetivos	El objetivo principal de este caso de uso es el de permitir al usuario visualizar los diferentes tipos de gráfica creados por el sistema para los datos introducidos por el mismo en el sistema
Requisitos asociados	Requisito 18-Requisito 22-Requisito 23
Descripción	Este caso de uso mostrará cómo es el proceso de visualización de las gráficas generadas en el sistema a partir de los datos pre registrados y almacenados en nuestro sistema

Precondición	Las necesidades que se deben cumplir para este caso de uso es la de haber iniciado la aplicación, que se haya creado con anterioridad un usuario utilitario de la misma y haber realizado la inserción de un registro glucémico sobre la que se quiera realizar la eliminación. En caso de no disponer de ninguno la gráfica no se generará.	
Secuencia Normal	Paso	Acción
	1	El usuario inicia desde el sistema operativo la aplicación
	2	El sistema mostrará la pantalla principal de navegación al usuario tras su inicio
	3	El usuario pulsará sobre el botón identificado por el icono de Gráficas
	4	El sistema reconducirá al usuario a la nueva pantalla de Gráficas, donde se dispondrá el espacio reservado para la definición de las mismas
	5	El usuario esperará a que se realice la animación de carga de las gráficas
	6	El sistema mostrará dos casillas seleccionables, que determinarán el tipo de gráfica que quiere visualizarse
	7	El sistema mostrará las gráficas durante el tiempo deseado por el usuario
Postcondición	Los datos almacenados en la base de datos, pasan a visualizarse de una manera gráfica, de tipo pastel o lineal, dependiendo de la opción seleccionada	
Excepciones	Paso	Acción

	4	E.1- En caso de que se produzca algún error al conectar con la base de datos, esta devolverá un mensaje de aviso al usuario para que realice un nuevo intento pasado un tiempo
Rendimiento	Paso	Cota de tiempo
	4	1-3 segundos
	6	1-2 segundos
Frecuencia esperada	Sin frecuencia previsible	
Comentarios	Sin comentario adicionales	

Tabla 40: Caso de uso 6

3.6.7-Caso de uso 7: Realizar recuperación de seguridad de los datos respaldados en Google Drive

CU-07	Recuperación de los datos que han sido almacenados por la copia de seguridad automática del sistema	
Objetivos	El objetivo principal de este caso de uso es el de permitir al usuario obtener los datos de la copia de seguridad realizada de manera automática por el sistema, en caso de pérdida o necesidad	
Requisitos asociados	Requisito 24-Requisito 25	
Descripción	Este caso de uso mostrará cómo es el proceso de recuperación de una copia de seguridad de los datos del dispositivo	
Precondición	Las necesidades que se deben cumplir para este caso de uso es la de haber iniciado la aplicación, que se haya creado con anterioridad un usuario utilitario de la misma y que se haya producido al menos la realización de una copia de seguridad de nuestros datos	
Secuencia Normal	Paso	Acción
	1	El usuario inicia desde el sistema operativo la aplicación
	2	El sistema mostrará la pantalla principal de navegación al usuario tras su inicio

	3	El usuario pulsará sobre el botón identificado por el icono de Copia De Seguridad, ubicado en la parte superior derecha de dicha pantalla
	4	El sistema reconducirá al usuario a la nueva pantalla de inicio de sesión de Google
	5	El usuario esperará a que se realice verificación de su usuario y contraseña en Google Drive
	6	El sistema mostrará la ubicación de la carpeta personal en Google Drive, donde aparecerá el archivo del sistema que ha sido respaldado
	7	El sistema devolverá al usuario a la página de inicio una vez seleccionado el archivo que se pretende
Postcondición	Los datos almacenados del sistema en el almacenamiento local, pasarán a sobrescribirse con los almacenados en la nube	
Excepciones	Paso	Acción
	5	E.1- En caso de que se produzca algún error al conectar con el servicio de Google Drive, este devolverá un mensaje de aviso al usuario para que realice un nuevo intento pasado un tiempo
	6	E.2 – En caso de que el sistema no pueda descargar el archivo almacenado en la nube, este devolverá al usuario un mensaje para que trate de realizarlo pasado un tiempo, y que compruebe su conexión a internet
Rendimiento	Paso	Cota de tiempo
	4	1-2 segundos

	5	1-5 segundos
	7	5 segundos
Frecuencia esperada	Sin frecuencia previsible	
Comentarios	Sin comentario adicionales	

Tabla 41: Caso de uso 7

3.6.8- Diagrama de casos de uso

Durante el apartado anterior, hemos definido un conjunto de casos de uso de nuestro sistema que permiten definir la interacción del mismo con el usuario, mostrando además un conjunto de opciones y funcionalidades que este ofrece conforme a los requisitos dados. Esta abstracción como se ha comentado anteriormente, permitirá a los desarrolladores conocer en mayor medida cómo deben desarrollarse los diferentes aspectos de nuestro sistema, y cómo estos interactuarán con el usuario final.

Pero para completar esto, es necesaria la definición de un diagrama explicativo capaz de aclarar aún más si cabe nuestra interacción usuario-sistema. Por ello, hemos definido en la página siguiente dicho diagrama:

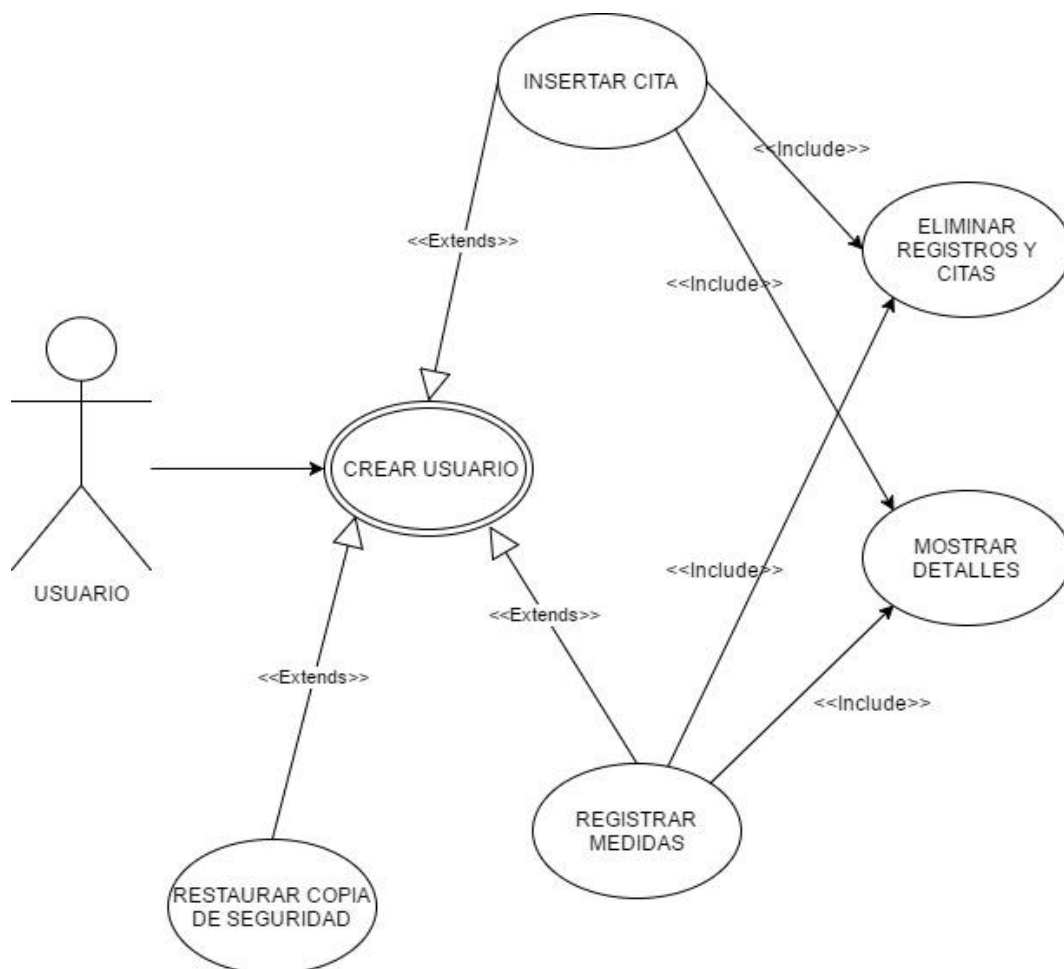


Ilustración 15: Diagrama de casos de uso

En el anterior diagrama, se muestran todos los casos de uso y sus relaciones entre sí, de tal manera que se categorizan con la notación de inclusión o extensión (<<include>> o <<extend>> en el diagrama). Los casos de uso que se relacionan mediante la inclusión, hacen referencia a aquellos que son dependientes entre sí, es decir que sin el caso de uso al que hace referencia, este no podría desarrollarse. Por otro lado, se definen casos de uso relacionados entre sí mediante la extensión, lo que denotará que dicho caso de uso es una funcionalidad que extiende al que referencia.

4-Diseño del sistema

4.1- Introducción al modelo 4+1

Así como en los apartados anteriores nos hemos centrado en definir de manera lo más clara y concisa nuestros sistemas software, tratando de aportar un conjunto de requisitos y casos de uso que permitan reproducirlo al más bajo nivel de la programación, a continuación, se presentarán los elementos que nos permitirán realizar un nuevo proceso de desarrollo: el diseño.

Existen gran cantidad de metodologías y formas de expresar un diseño software, pero no todas las posibles facilitan al programador la tarea de transcribir lo deseado a lo implementado. Tratando de evitar el mayor número de errores en este proceso de transcripción que se produce en esta parte del ciclo de vida de un desarrollo software, hemos tratado de conjugar nuestro análisis con un diseño dado por la metodología de desarrollo software denominada como 4+1. [30]

De esta manera el software según esta metodología, se centra en una definición de sus elementos, relaciones y funciones dentro del sistema, basada en diferentes puntos de vista o niveles de abstracción. Por ello esta es una metodología completa que permitirá una precisión alta, al tratar un mismo desarrollo desde diferentes niveles, estudiando cada uno de sus puntos y haciéndolo más claro y mejor definido, permitiendo crear una arquitectura de mayor calidad y eficiencia.

Este conjunto de vistas se desarrolla en las siguientes:

- **Vista Lógica:** Consiste en una abstracción del más alto nivel que nos permitirá conocer las funcionalidades que el sistema puede aportar al usuario. Esta vista está caracterizada por el diagrama de clases y el diagrama de componentes.
- **Vista Física:** Esta vista consiste en la abstracción del sistema hasta el punto de conocer cómo se relaciona nuestro software en una red global, además de su relación con otros sistemas interconectados. Responde a la pregunta de *“¿Cómo interacciona mi software con el entorno global?”*.
- **Vista de Procesos:** Esta vista nos permite conocer cómo se interrelacionan los procesos de nuestro sistema, teniendo en cuenta normalmente requisitos no funcionales. Esta vista se describirá como un conjunto de diagramas de secuencia, que permitirán conocer desde un nivel más bajo, algunos procesos que se desarrollan al utilizar nuestro sistema.
- **Vista de Desarrollo:** Consiste en organizar la modularidad de nuestro software, utilizando diagramas habitualmente de paquetes. Debido a la sencillez de nuestro sistema, esta vista no aportará una gran cantidad de información por lo que se ha decidido excluirla del modelo 4+1 y realizar dicha adaptación al modelo.
- **Vista de escenarios:** Esta vista consiste en una conjunción o combinación de todas las anteriores, donde se desarrollan los diferentes casos uso de manera que se cumplan todos los requerimientos y relaciones que se han especificado. Dado que el diagrama de casos de uso ha sido desarrollado en

el apartado anterior, siempre que sea necesaria la definición o referencia de dicha vista, recurriremos a este apartado.

4.2- Vista lógica

Ya hemos dado una pequeña definición de lo que implica el desarrollo de una vista lógica en la arquitectura de nuestro software, pero en este apartado vamos a profundizar un punto más en su desarrollo, así como aplicar dicha teoría para crear una parte de nuestra solución de diseño.

Nuestra solución se basará en dos tipos principales de diagramas: **el diagrama de componentes y el diagrama de clases**. Este conjunto será capaz de soportar los requisitos funcionales determinando los servicios que están enfocados a la utilización por parte de los usuarios finales.

Pero aún no hemos determinado como vamos a englobar los componentes que conforman nuestro sistema, en abstracciones de mayor grado capaces de mostrar la realidad de nuestra funcionalidad sin contener lenguaje de bajo nivel. Para realizar esto, nuestro sistema de bajo nivel estará agrupado en componentes, entre los cuales tendremos los siguientes tipos:

- **Componente de Interfaz y pantallas**, que será el encargado de contener todas aquellas clases referentes a la interfaz gráfica y de diseño de nuestra aplicación, y que no se puedan definir como funcionalidades de negocio o clases de entidad. Estas tienen una gran importancia en nuestro sistema, pero son totalmente modificables por parte de los desarrolladores siempre que respeten las restricciones propias del sistema.
- **Componente de Entidades y objetos**, que será el encargado de contener todas aquellas clases que tengan una independencia por si solas, siendo clases con las cuales se interactúa en nuestro sistema de manera necesaria para poder realizar diferentes tipos de funciones.
- **Componente de Operaciones**, que será el encargado de disponer todas aquellas clases que sean necesariamente funcionalidades del sistema, utilizando otras clases para generar resultados o funciones clave en nuestro sistema. Son los componentes con la funcionalidad más práctica.
- **Componente de Base de Datos**, que será el módulo encargado de contener a aquellas clases o clases de nuestro sistema que guarden relación con la existencia de la base de datos.

Este conjunto como se ha definido, permitirá conseguir una abstracción de las dependencias lógicas que engloban a nuestro sistema, y al ser elementos físicos

de nuestro sistema, dentro podrán contener interfaces, clases etc. Por este motivo, los diagramas de componentes como veremos a continuación, están muy relacionados con los diagramas de clases.

Tras esta definición formal de los que nuestros componentes han de contener, vamos a tratar de definir cada uno de ellos por separado, tratando de explicar qué contendrá cada uno de los componentes referido a nuestro sistema particular.

Los componentes de interfaz, como hemos denominado antes, se centran en las clases que contienen un carácter más gráfico y visual de nuestro sistema. Estas se conformarán por:

- **Graficos:** Será la clase en la cual se genera el activity para el menú de gráficas de nuestra aplicación, que nos permitirá realizar una visualización de los registros a modo de abstracción para los registros introducidos por el usuario.
- **PaginaPrincipal:** Consiste en la página de bienvenida que cargará el sistema al iniciarse, esta funcionará a modo de punto común para navegar a los demás menús.
- **Perfil:** Será la clase en la cual se generará el activity que mostrará el menú de usuario, navegando a través del sistema para llegar a ella.
- **Registros:** Será la página en la cual se puedan visualizar los registros del usuario introducidos en el sistema.

Por otro lado, tenemos el componente de Entidades, cuya definición nos dicta que se dispondrán en él clases cuya independencia en el sistema hará que sirvan de apoyo para las funcionalidades del paquete de negocio y de base de datos. En él se contendrán las siguientes clases:

- **Cita:** Será la clase que generara los objetos del tipo Cita, en los cuales se almacenarán datos como el doctor, hospital, fechaCita, horaCita, descripcionCita. Estos definirán los campos de nuestras citas generadas por el usuario, y almacenadas en la base de datos posteriormente.
- **FichaCitas:** En esta clase se generarán las fichas de la cita que más tarde serán visualizadas en nuestro sistema por parte de los usuarios, a modo de listas deslizables.
- **Persona:** En esta clase se contendrán todos los datos correspondientes a los objetos del tipo Persona, tales como el nombre, apellidos, edad y unidadesIDiarias. Estos objetos serán almacenados más tarde en la base de datos.
- **Medida:** Será la clase que generara los objetos del tipo Medida, en los cuales se almacenarán datos como glucemia, horaMedida, fechaMedida

y descripcionMedida. Estos definirán los campos de nuestras medidas generadas por el usuario, y almacenadas en la base de datos posteriormente.

- **FichaMedidas:** En esta clase se generarán las fichas de las medidas que más tarde serán visualizadas en nuestro sistema por parte de los usuarios, a modo de listas deslizables.

Como tercer paquete, se ha definido el paquete relativo a las funcionalidades de negocio. En este se incluirán los componentes capaces de realizar las funcionalidades de nuestro sistema, generando resultados directamente asociados a la interacción con el usuario. En este paquete se definirá una única clase, denominada **Operaciones** que contendrá los métodos que siguen:

- **NuevaMedida:** Este método se desarrolla con una funcionalidad clara, que es insertar y obtener las medidas que los usuarios han introducido de manera continuada durante un periodo de tiempo. Está directamente relacionada con los elementos de la interfaz gráfica, las entidades y la base de datos.
- **AñadirMedidaLista:** Cuya función principal será la de añadir las nuevas medidas a la lista que las contenga, para simplificar el trabajo y la organización.
- **NuevaCita:** Este método se desarrolla al igual que en el caso anterior, con la funcionalidad de insertar y obtener las medidas que los usuarios requieran. De igual manera está muy relacionada con la interfaz, las entidades y la base de datos del sistema.
- **AñadirCitaLista:** Cuya función principal será la de añadir nuevas citas a la lista que las contenga, para simplificar el trabajo y la organización.
- **ObtenerGrafica:** Este método al contrario que las otras dos, se centra en reproducir de manera gráfica las diferentes medidas del usuario, siendo capaz de reproducir gráficas de diferentes tipos.
- **GenerarBackup:** Este método será la encargada de generar las copias de seguridad en nuestro sistema en la nube, según los datos existentes en nuestro sistema, y de manera automática.
- **RecuperarBackup:** Este método será el encargado de obtener los datos que el sistema ha almacenado con anterioridad en la nube, en el momento en el que el usuario lo necesite.

Por último, tendremos un componente final que únicamente será el encargado de contener una clase. Este es el componente de Base de Datos y será el encargado de almacenar los datos del sistema que el usuario pretenda registrar. En él se incluirá la clase **SQLTables**, cuya función en el sistema es la de crear, generar y mantener la base de datos que utilizaremos como medio de almacenamiento, activa y estable.

Por consiguiente, y tras la definición de todas las clases propias para cada uno de los componentes, obtendremos la definición única de un diagrama de los mismos, en cuyo interior como se ha explicado, se contendrán cada una de las clases descritas.

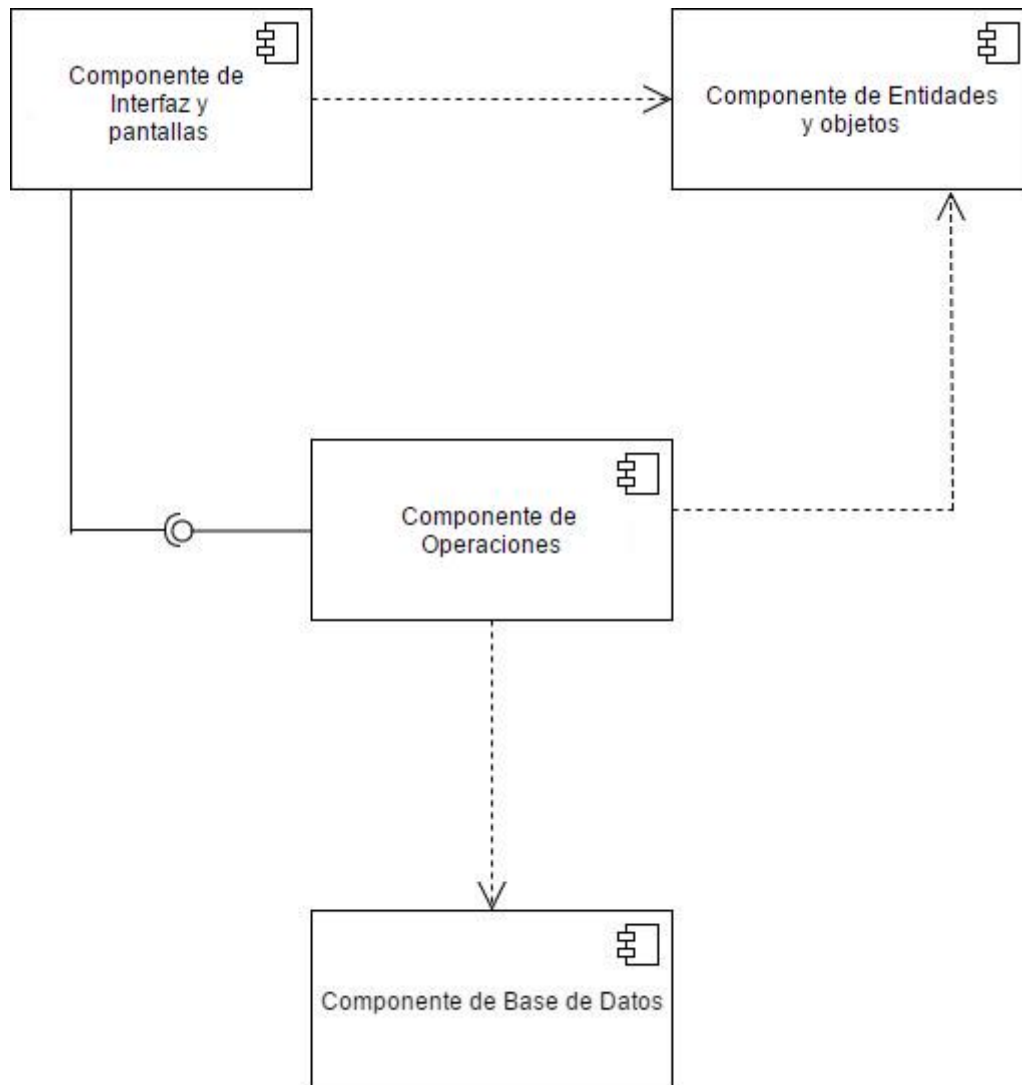


Ilustración 16: Diagrama de componentes del sistema

En dicho diagrama podremos observar las interrelaciones entre los componentes de nuestro sistema, observando que como es destacable, el componente de negocio que es el encargado de realizar todas las operaciones y determinar de manera dinámica cual será la interfaz, depende del componente de entidades y bases de datos, que es el que le proporcionará los datos para poder operar. Por otro lado, el componente de interfaz necesita del componente de negocio para poder reproducir los datos que el usuario necesita visualizar, variando en función de los cálculos realizados.

Por último, los datos que están almacenados en la base de datos, se obtienen mediante las funciones almacenadas en el componente de negocio de igual manera, y serán accedidas o insertadas por este.

Pero como acabamos de comentar, este diagrama únicamente muestra a un muy alto nivel cual es la solución de nuestro sistema, y pueden producirse ambigüedades entre las diferentes clases que lo conforman a dicho nivel de abstracción. Y es por esto por lo que los desarrolladores necesitan tener un punto de vista más preciso de lo que al sistema se refiere. Esta nueva vista se obtendrá mediante el diagrama de clases. Dicho diagrama nos ofrecerá una visión en profundidad de las interacciones, dependencias y relaciones entre las clases que conforman cada uno de nuestros componentes.

Este diagrama se dispondrá en la siguiente página, mostrando todas nuestras clases, así como su composición interna y de relación:

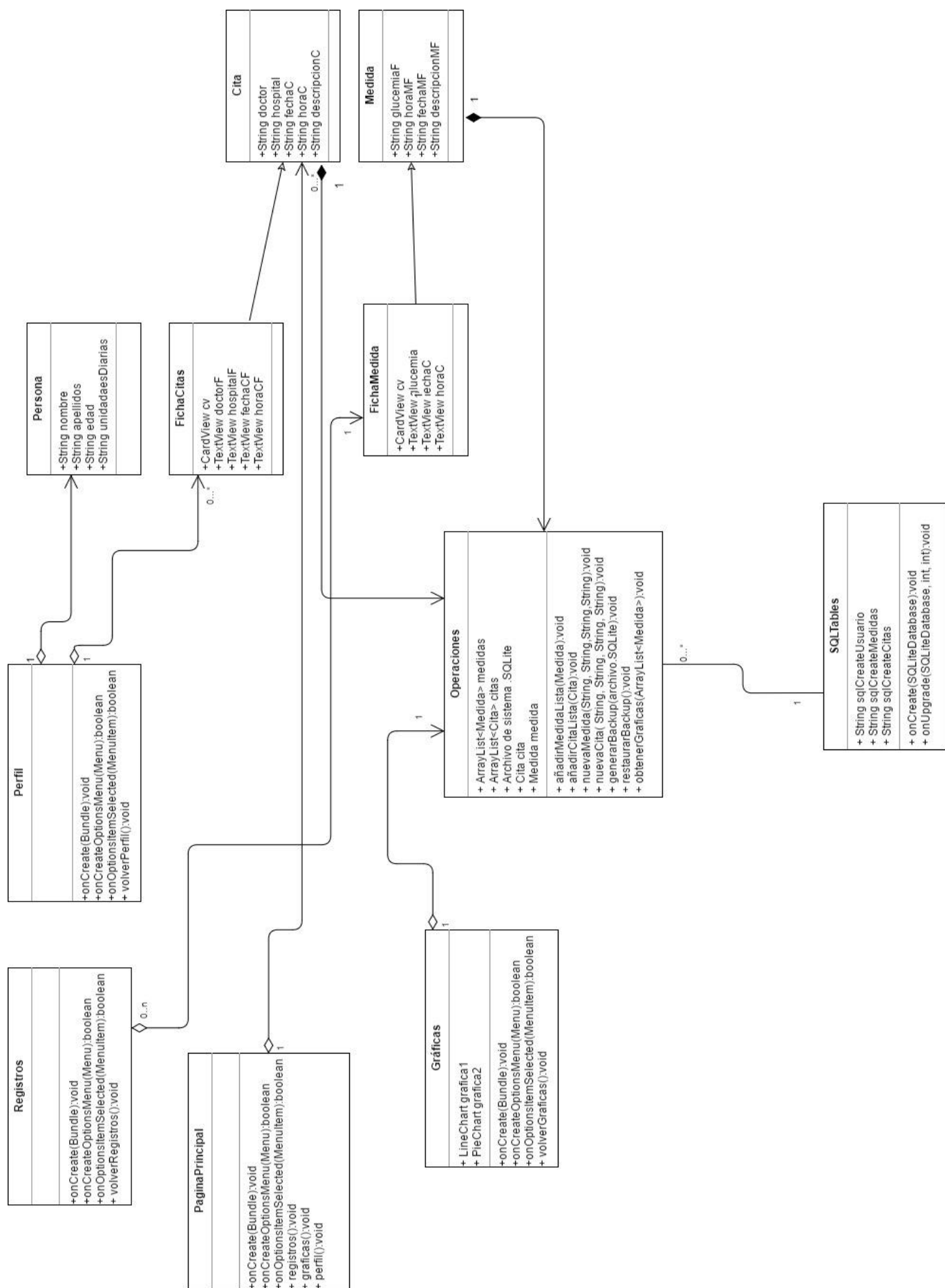


Ilustración 17: Diagrama de clases del sistema

4.3- Vista física o de despliegue

Hemos comentado durante los apartados anteriores la función de nuestro modelo de vistas 4+1, cuya intención se basaba en tratar de mostrar desde todas las posibles visualizaciones de nuestro sistema, como este interactuaba con el entorno y consigo mismo. De esta manera, el siguiente nivel de abstracción que toca definir es el que muestra la escalabilidad del sistema y su visión hacia sistemas externos al propio. Esta visión es la denominada visión de despliegue o física.

Para la creación de esta vista, en nuestro caso y dado que el sistema es un sistema que trabajará de manera aislada a otros, tendrá un diagrama bastante simplificado, que combinará el diagrama de componentes de nuestro sistema, con otros sistemas interconectados. Nuestro sistema como se ha dicho se situará en un nodo propio, que hará referencia a la ejecución en nuestro dispositivo privado y particular, ya que este es uno de los puntos sobre los cuales se enfoca la utilidad del mismo. De igual manera, la conexión a la base de datos y al servicio de copias de seguridad en la nube, se encontrarán corriendo en otras máquinas independientes a las propias, y eso es lo que se intentará mostrar en dicho diagrama.

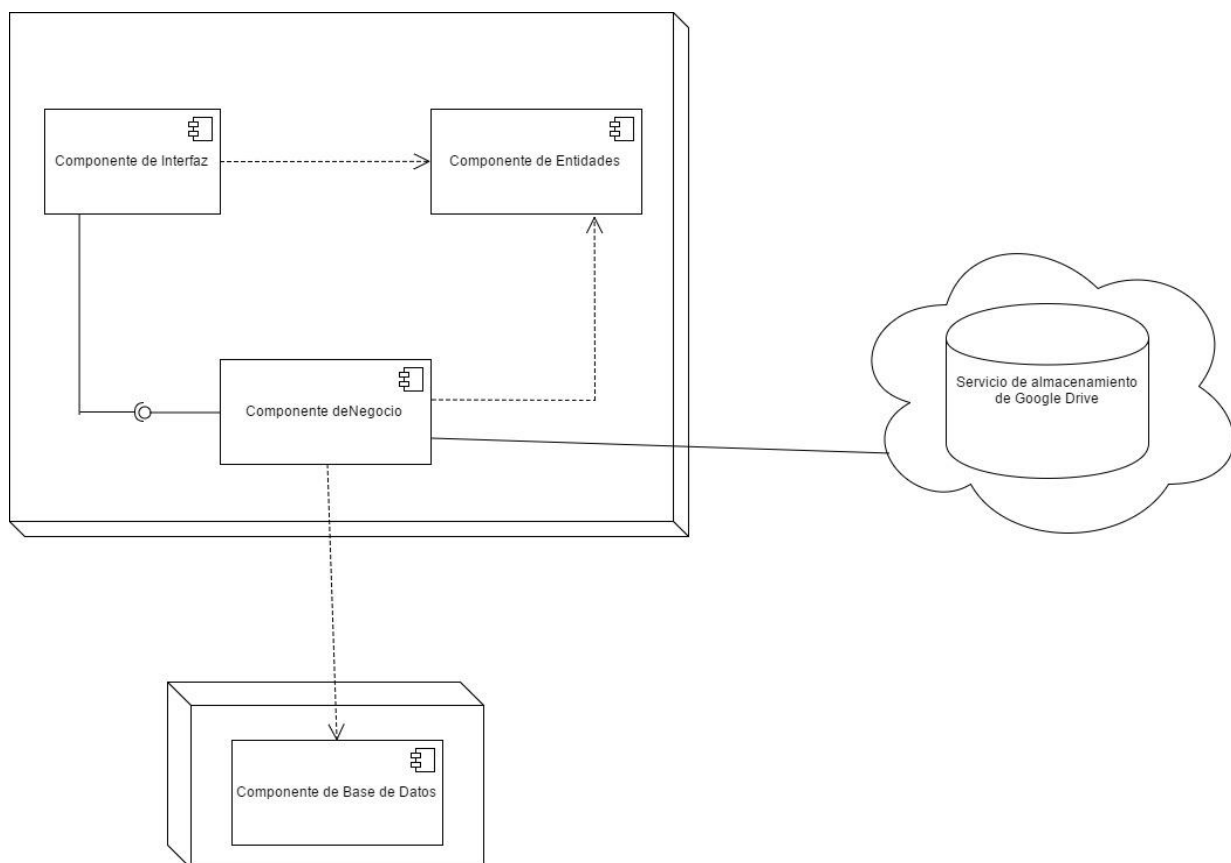


Ilustración 18: Diagrama de despliegue del sistema

La interpretación de este diagrama se centra en las tres necesidades básicas que posee nuestro sistema. La primera y más importante es la de funcionar como un dispositivo de registro y control de valores glucémicos en nuestro dispositivo móvil. Esto a modo de sistema contendrá en un único nodo todas las funcionalidades necesarias para que sea nuestro propio dispositivo el único necesario para almacenar datos en forma de ficheros en memoria, por lo que este es el motivo por el cual un único nodo puede contener todos los componentes de interfaz, negocio y entidades, ya que son los necesarios para cumplir con dicha funcionalidad.

Por otro lado, existe el caso en el que el usuario pretenda generar una copia de seguridad o restaurar de igual manera los datos en el sistema. Este caso necesitará de un servicio autónomo proporcionado por el proveedor del servicio de cloud en el que se almacenarán los datos, por lo que se encuentra aislado del nodo del sistema, pero interrelacionado con él.

Por último, existe la funcionalidad de realizar una gestión, organización y almacén de datos del usuario. Esto se obtendrá mediante un servicio de bases de datos, que, al estar proporcionado de manera externa, este también debe estar aislado del sistema y dispuesto fuera de nuestro nodo de ejecución.

Para este proceso de desarrollo no es necesaria más de una máquina para definir un sistema completo y capaz, motivo por el cual el diagrama únicamente cuenta con un nodo de ejecución principal sobre el que se extienden otras funcionalidades. Por este motivo si han de ser necesarias las colaboraciones con servicios de bases de datos y servicios de cloud capaces de satisfacer al sistema.

4.4- Vista de procesos

La tercera parte de nuestro modelo se centra en el cumplimiento de los requisitos especificados durante los apartados anteriores. En esta vista se trata de ofrecer el conjunto de procesos y como estos interactúan durante la ejecución de diferentes tipos de acciones en nuestro sistema. Este apartado vendrá claramente definido por los diagramas de secuencia, los cuales nos ayudarán a entender cómo funciona en el tiempo nuestro sistema, tras una ejecución del usuario. Esto permitirá obtener una visión temporal y cronológica de cómo se desarrollan dichas funcionalidades, precisando a nivel de método en la visualización.

Además de esto, este diagrama o definición de vista está claramente relacionada con el diagrama de componentes anteriormente definido, pues nos permitirá conocer de manera más concreta como se distribuyen las tareas que debe realizar nuestro sistema, entre los diferentes componentes.

Para este tipo de diagramas, y debido a la sencillez estructural que engloba a nuestra aplicación, se ha decidido implementar dichos diagramas a modo explicativo sobre varios de los casos de usos más completos que se han definido. Este conjunto de casos de uso serán la introducción de medidas por parte del usuario en el sistema, y almacenándose estas en la base de datos, y la creación de gráficas.

De esta manera, el caso de uso que se va a realizar primero se centra en la inserción de una medida glucémica por el usuario en el sistema. El diagrama correspondiente será el que prosigue:

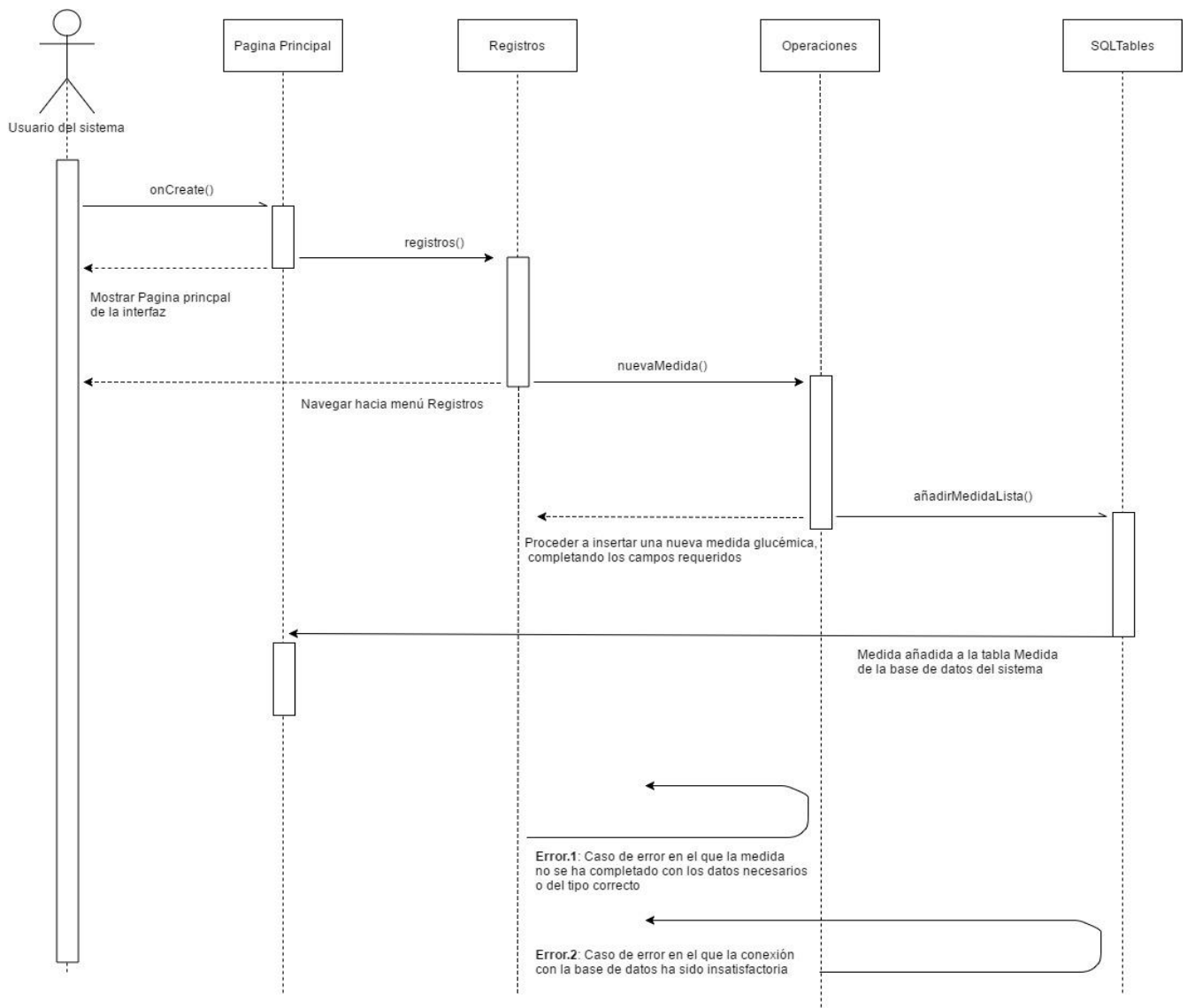


Ilustración 19: Diagrama de secuencia (I)

Como puede observarse, el diagrama muestra como un usuario es capaz de acceder a la clase **Registros**, donde se deberán completar un conjunto de campos para llamar a los métodos de inserción, propios de la clase **Operaciones**, que de manera automática insertarán los valores obtenidos en la base de datos determinada por la clase **SQLTables**, devolviendo al usuario al menú de inicio, tras realizar este proceso.

Como segundo diagrama de secuencia, se tomará como se ha introducido el caso de uso referente a la generación de gráficas personales sobre los datos introducidos por el usuario, siguiendo los pasos del anterior diagrama, en el sistema. De esta manera

la clase **Graficas** obtendrá las definiciones de graficas de varios tipos, lineales y de pastel, para mostrar al usuario. El diagrama de secuencia asociado a este caso de uso se define como:

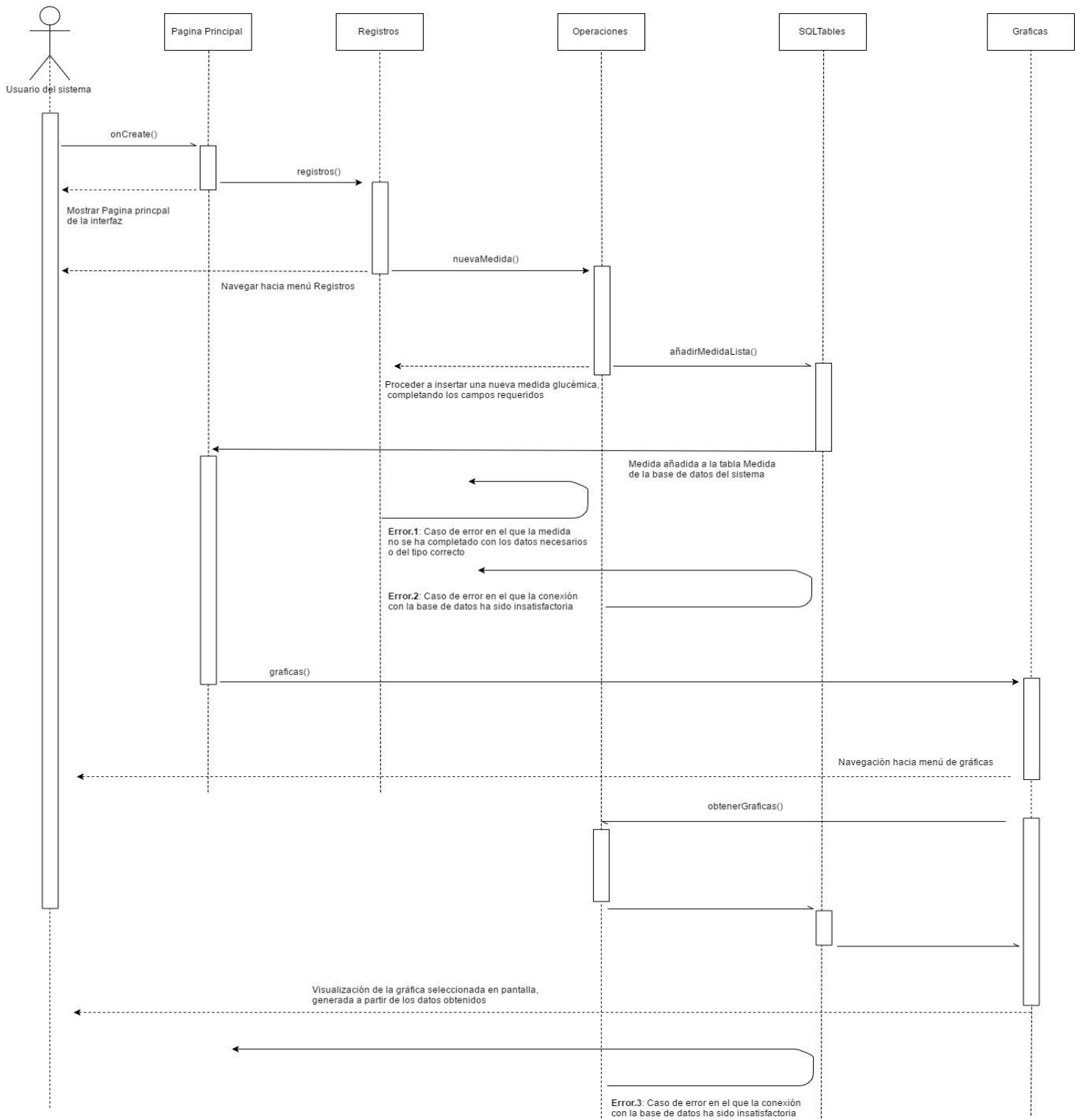


Ilustración 20: Diagrama de secuencia (II)

4.5- Mapa de pantallas

Para el desarrollo de nuestro proyecto, como se ha dicho anteriormente, hemos decidido no implementar la vista de desarrollo, por lo que se tratará otro aspecto importante a la hora de diseñar un software de tipo aplicación: la navegación entre pantallas.

La navegación entre pantallas se centra en definir mediante un mapa de pantallas, las diferentes interfaces o capacidades gráficas de las que va a disponer nuestro usuario utilitario para poder acceder a las diferentes secciones de nuestro sistema. Este mapa es de gran utilidad pues puede ayudar a entender cuáles son las dimensiones de nuestra aplicación, a mejorar su funcionamiento y utilidad, así como a mejorar la curva de aprendizaje de un usuario, aprendiendo los pasos que le permitirán realizar los objetivos que pretenda con la aplicación.

En nuestro caso nuestro sistema se compondrá de las pantallas que hemos podido definir durante el documento, por ejemplo, en el diagrama de clases. A modo de resumen estas se centrarán en:

- Pantalla de página principal
- Pantalla de Perfil
- Pantalla de Registros
- Pantalla de Gráficas
- Pantalla de adición de citas
- Pantalla de notas sobre las citas almacenadas
- Pantalla de adición de registros
- Pantalla de notas sobre las medidas almacenadas
- Pantalla de restauración de archivo de copia de seguridad
- Pantalla de ayuda
- Pantalla de tutorial

Todas ellas estarán representadas en nuestro sistema como un grafo, en el que cada nodo se dispondrá como una pantalla, partiendo de la inicial que se produce al iniciar el sistema, que es la pantalla de Página principal, y mostrando como aristas del grafo la navegación entre ellas. Cabe destacar que la pantalla de tutorial únicamente se inicializa una vez en el sistema, siendo esta la primera vez que el usuario inicia el mismo, sin existir ningún usuario registrado. Además, las pantallas de ayuda se dispondrán en el momento en el que el usuario pulse sobre alguno de los botones situados para requerir dicha ayuda o información.

El mapa de pantallas de nuestro sistema, por tanto, lucirá como sigue:

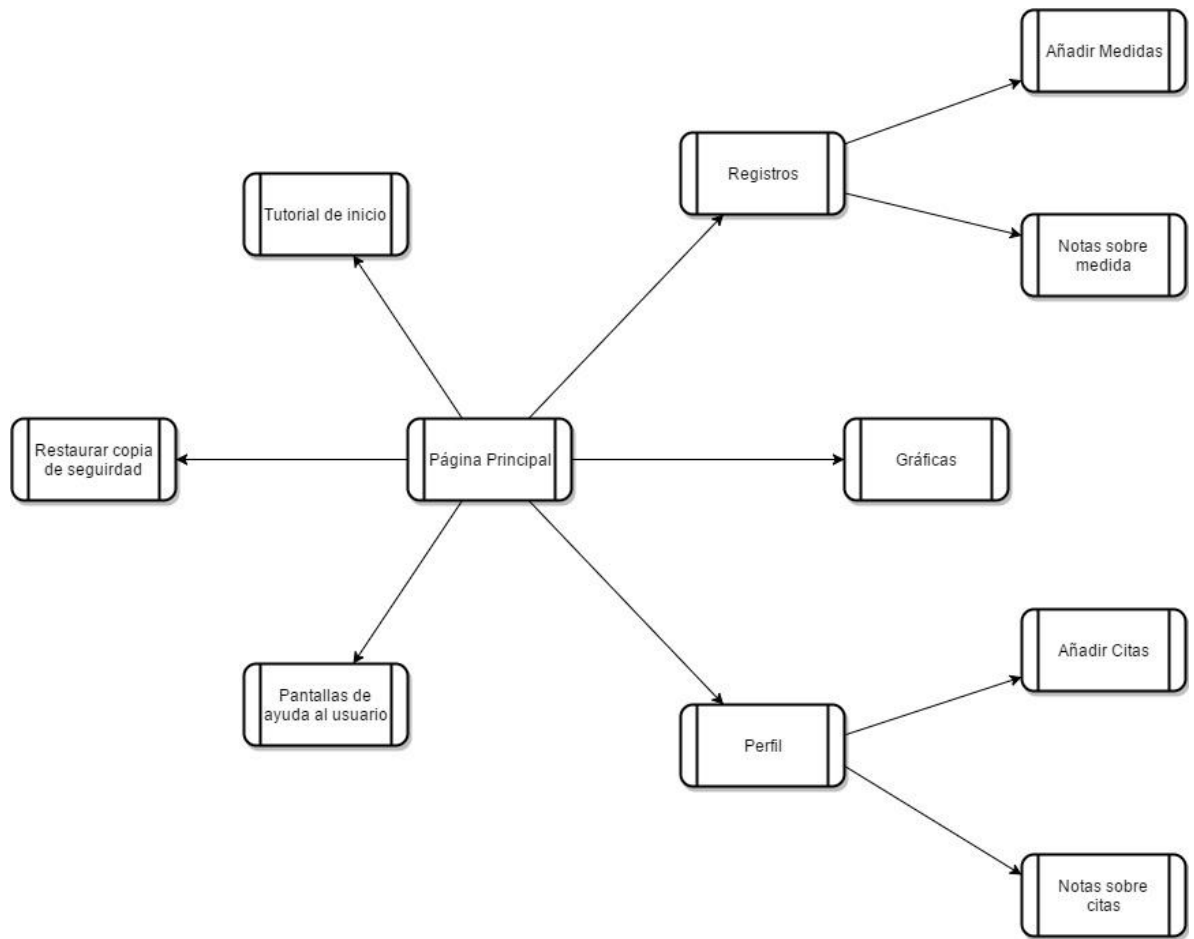


Ilustración 21: Mapa de pantallas del sistema

4.6-Entorno tecnológico

4.6.1- Desarrollo en Android

Durante el documento, se estudió Android en su aspecto más general, y como su propia definición dicta, un sistema operativo con una historia de desarrollo y en continua actualización. Pero esto no podría llevarse a cabo sin un sistema de desarrolladores capaz de implementar todos los cambios que dicho sistema requiere, adaptando cada una de las características a los requerimientos necesarios, tanto por parte de la industria, como por parte de los usuarios de a pie. Es por este motivo por el cual vamos a definir cuáles son los aspectos esenciales del desarrollo en Android, desde el entorno, hasta los aspectos básicos de la programación de este tipo de lenguaje.

Este análisis nos ayudará a entender cuál ha sido nuestro enfoque a la hora de desarrollar el proyecto, y hacer entender al usuario las bondades que puede ofrecer este lenguaje.

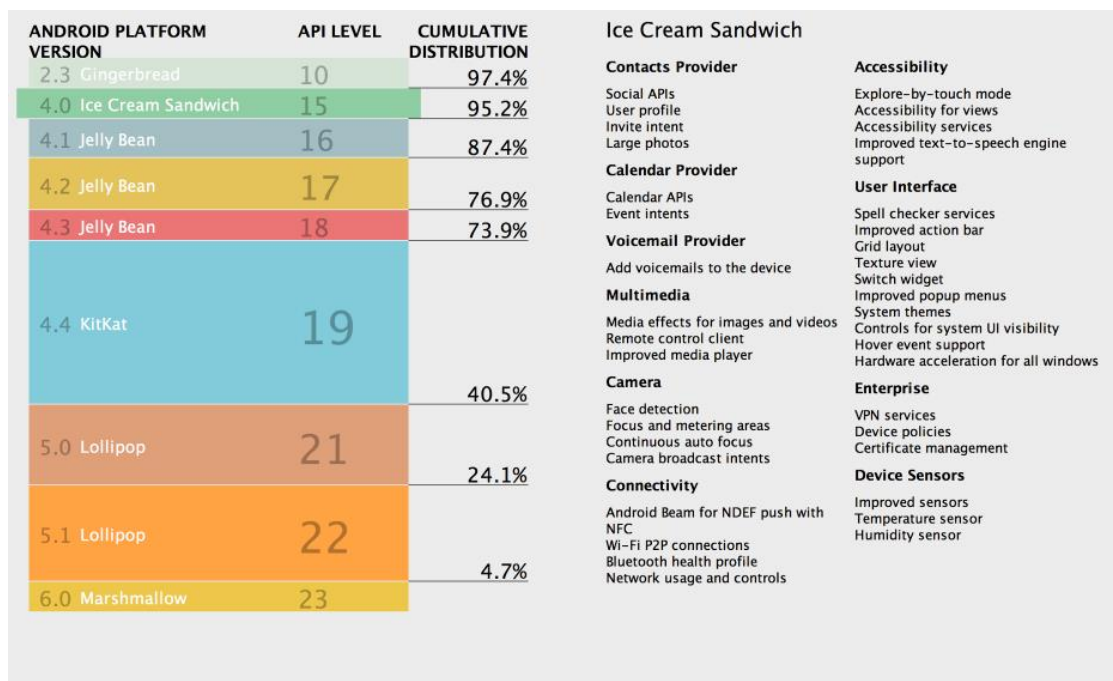


Ilustración 22: Distribución del desarrollo en Android por versión [31]

4.6.2- El entorno de programación de Android: Android Studio

“Android Studio es el IDE oficial de Android. Está diseñado para que Android pueda acelerar el desarrollo y te permita crear las apps de mejor calidad para todos los dispositivos de Android.

Ofrece herramientas personalizadas para programadores de Android. Se incluyen herramientas completas de edición, depuración, pruebas y perfilamiento de códigos.”

[32]

Esta es la definición oficial de este entorno de programación, que ha evolucionado hasta generar un programa propio y multiplataforma, evitando ahora el uso de programas de terceros como **Eclipse** o **NetBeans**. Este hecho ha generado que puedan introducirse mejoras basadas en **IntelliJIDEA**, para mejorar el procesamiento de los fragmentos de código. Estas mejoras se traducen en un entorno inteligente capaz de generar cambios en el código sin necesidad de recompilar un proyecto completo, algo que puede marcar una gran diferencia, en el caso de tener que generar de nuevo un sistema demasiado complejo y con grandes niveles de interacción. [33]

Por otro lado, el editor de este entorno es capaz de reconocer código, así como su funcionalidad dentro de nuestro programa, para poder sugerir modificaciones a la hora de desarrollar, y permitir al programador centrarse en la tarea de desarrollo, y no en la optimización o corrección de errores.

Además de esta plataforma ofrece un sistema de emuladores capaz de generar una virtualización en tiempo real de nuestra aplicación, es decir, que permite al desarrollador poder interactuar con una pequeña máquina virtual, para poder visualizar los cambios en su código, e interactuar con estos en primera persona, como si se tratara de cualquier tipo de dispositivo móvil real, incluyendo además la adaptación a diferentes tipos de pantalla. Este aspecto es uno de los más completos de la plataforma, ya que se puede interactuar con cualquier tipo de característica requerida por el sistema, y utilizada en dispositivos reales, pero de manera virtual.

Para trabajar con este entorno, el lenguaje de programación utilizado, combina tanto una parte de funcionalidad e implementación de funcionalidades basada en **Java**, como una parte de diseño de interfaz e interacción basado en lenguaje **XML**. Esta combinación proporcionará una serie de ventajas para el desarrollador, tales como la gran cantidad de librerías y la gran adaptación que tiene para el desarrollo de aplicaciones móviles, cubriendo un gran déficit de otros sistemas. Además, la utilización de XML, hace que se puedan generar módulos software, permitiendo conocer las piezas por separado que componen el mismo.

Esto a su vez está potenciado con un diseñador propio, que permitirá mediante una interfaz gráfica y sencilla la creación de diferentes pantallas o actividades propias de la aplicación.

Además de esto, se proporciona al usuario una interfaz amigable, complementada con modos de autocompletar, e incluso servicios de generación de archivos **APK** o servicios Cloud de Google.

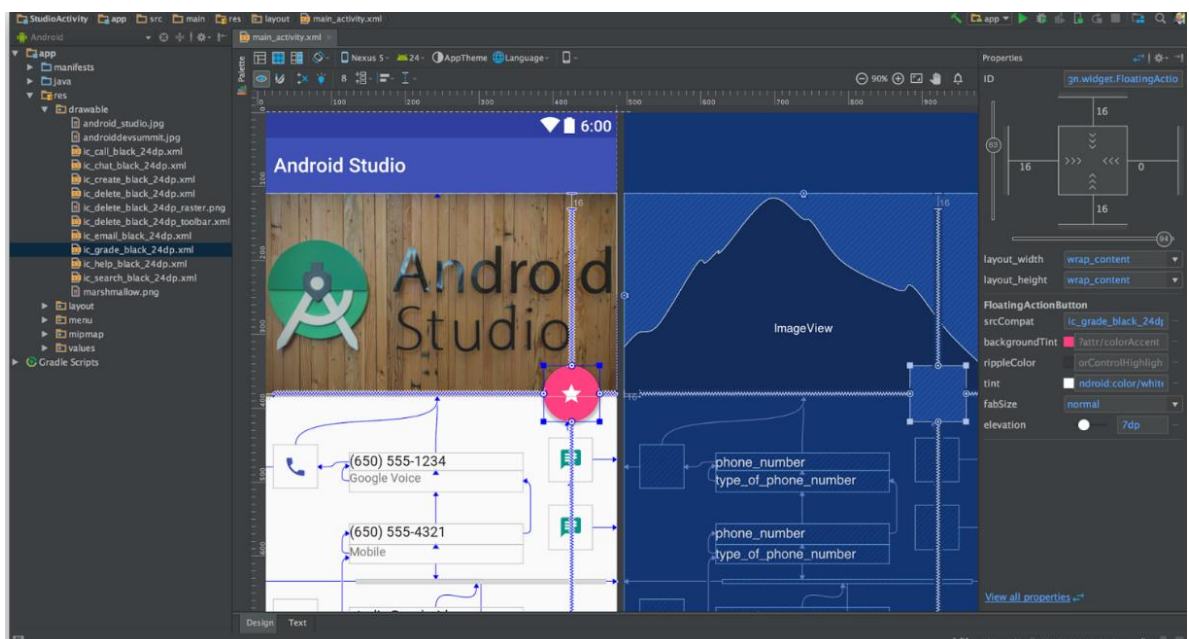


Ilustración 23: Android Studio [34]

4.7- Implementación

Durante el apartado anterior, nos hemos centrado en crear una definición clara y concisa de la organización arquitectónica de nuestro sistema, esto se centra en la definición de un conjunto de componentes, de funciones y métodos, que, funcionando en conjunción, generan un conjunto de salidas útiles para el usuario. Pero el nivel de abstracción mantenido durante todo este apartado no nos ha permitido adquirir ciertos aspectos relacionados con la implementación real de nuestro sistema. Eso es lo que trataremos de desarrollar durante este apartado, el alcanzar un nivel de abstracción tan preciso que nos permita ver detalles del código que hemos generado para producir este sistema software.

A continuación, en este apartado, definiremos un conjunto de pequeñas partes de nuestro código, susceptibles de ser analizadas, dadas sus características o la necesidad que estos tienen en el sistema. Este conjunto de decisiones durante la implementación, no dictan la única forma viable para el desarrollo de una aplicación funcional, pero se han considerado acertadas por ciertos motivos en compensación a sus desventajas.

De esta manera el primero de los casos en los que la implementación ha sido decisiva, se centra en la implementación de las listas que ofrecen al usuario la visualización de citas y de los registros glucémicos. En este caso, Android muestra una de sus evoluciones más notorias, obligando al desarrollador, en caso de pretender cumplir sus patrones de diseño, a utilizar para la implementación de listas el elemento **RecyclerView**. Este tipo para la implementación permite la creación de listas en forma de tarjetas adaptables al contenido de las mismas. Esto es una implementación que se adapta perfectamente a nuestras necesidades, pues podremos desarrollar nuestras tarjetas, organizando sus componentes de interfaz en una clase a parte, para después permitir que esta se añada de manera dinámica, dependiendo del contenido.

Pero además de esto, este tipo de implementación basado en tarjetas, nos dará otra ventaja sobre otras, y esta es la capacidad para mostrar detalles o eliminar citas de manera precisa y específica sobre el contenido de la lista. Esto puede desarrollarse gracias a que el modelo de tarjetas implementado en el **RecyclerView**, permite identificar cada elemento como único, en el momento de la creación de estos.

```
public class FichaCitasA extends RecyclerView.Adapter<FichaCitasA.FichaCitasVH> {  
  
    public static List<Cita> citas;  
  
    public static class FichaCitasVH extends RecyclerView.ViewHolder {  
        CardView cv;  
        TextView TVdoctor;  
        TextView TVhospital;  
        TextView TVfecha;  
        TextView TVhora;  
        TextView TVtituloglucemia;  
        TextView TVtitulohora;  
        TextView TVtitulofecha;  
  
        public FichaCitasVH(View itemView) {  
            super(itemView);  
            this.cv = (CardView) itemView.findViewById(R.id.cardCita);  
            this.TVdoctor = (TextView) itemView.findViewById(R.id.doctor);  
            this.TVhospital = (TextView) itemView.findViewById(R.id.hospital);  
  
            @Override  
            public FichaCitasA.FichaCitasVH onCreateViewHolder(ViewGroup parent, int viewType) {  
                View v = LayoutInflater.from(parent.getContext())  
                    .inflate(R.layout.filacita, parent, false);  
                FichaCitasA.FichaCitasVH pVH = new FichaCitasA.FichaCitasVH(v);  
                return pVH;  
            }  
  
            @Override  
            public void onBindViewHolder(final FichaCitasVH holder, final int position) {  
  
                holder.cv.setCardBackgroundColor(Color.rgb(132,195,190));  
  
                holder.TVdoctor.setText(String.valueOf(FichaCitasA.citas.get(position).getDoctor()));  
                holder.TVhospital.setText(String.valueOf(FichaCitasA.citas.get(position).getHospital()));  
                holder.TVfecha.setText(String.valueOf(FichaCitasA.citas.get(position).getFecha()));  
                holder.TVhora.setText(String.valueOf(FichaCitasA.citas.get(position).getHoraFicha()));  
                holder.TVtituloglucemia.setText(String.valueOf(FichaCitasA.citas.get(position).getGlucemia()));  
                holder.TVtitulohora.setText(String.valueOf(FichaCitasA.citas.get(position).getHoraGlucemia()));  
                holder.TVtitulofecha.setText(String.valueOf(FichaCitasA.citas.get(position).getFechaGlucemia()));  
            }  
        }  
    }  
  
    public FichaCitasA() {  
        citas = new ArrayList<>();  
    }  
  
    public void add(Cita cita) {  
        citas.add(cita);  
    }  
  
    public void remove(Cita cita) {  
        citas.remove(cita);  
    }  
  
    public int getItemCount() {  
        return citas.size();  
    }  
}
```

Ilustración 24: Fragmento de código 1

En este conjunto de imágenes puede observarse a grandes rasgos como es la secuencia de creación de un **RecyclerView** para las citas introducidas en la aplicación por el usuario. En él se deben definir los elementos visuales que contendrá nuestra ficha, para seguidamente llamar a los métodos **onCreateViewHolder()**, que reservará espacio en la lista para todos los elementos que se pretendan disponer, y **onBindViewHolder()**, que se encargará de rellenar las diferentes fichas con los valores que tendrá dinámicamente cada una de las tarjetas.

Pero otro de los aspectos importantes que tendremos en nuestros elementos **RecyclerView**, se basarán en la capacidad de los mismos para ser pulsados y, dependiendo de la duración de dicha pulsación, realizar una acción u otra. Esta implementación se llevará a cabo con varios métodos proporcionados por las librerías Android de manera nativa, denominados **onClickListener()** y **onLongClickListener()**. Dichos métodos crearán un **listener** para diferentes tipos de pulsación, que esperarán en el sistema en segundo plano hasta ser llamados, pero de manera constante.

En nuestro caso se han implementado dichos métodos para en caso de una pulsación simple, poder mostrar al usuario tanto las notas añadidas a una cita, como los detalles que el usuario pretenda añadir a una medición de su glucemia. A continuación, se muestran varias pantallas con la definición de los mismos en nuestro sistema:

```
View.OnClickListener c = new View.OnClickListener(){
    @Override
    public void onClick(final View v) {
        AlertDialog aviso = new AlertDialog.Builder(v.getContext()).create();
        aviso.setTitle("Notas sobre la cita");
        aviso.setMessage(FichaCitasA.citas.get(i).descripcionCita);
        aviso.show();
    }
};
holder.cv.setOnClickListener(c);
```

Ilustración 26: Fragmento de código 3

```
View.OnLongClickListener l1 = new View.OnLongClickListener() {
    Boolean borrado = false;

    @Override
    public boolean onLongClick(final View v) {
        AlertDialog.Builder confirmacionEliminar = new AlertDialog.Builder(v.getContext());
        confirmacionEliminar.setTitle("¿Está seguro de que desea eliminar la cita?");

        //Creamos los dialogos de borrar y cancelar
        confirmacionEliminar.setPositiveButton("Si", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                //Si pulsamos si, eliminamos la medida, en caso pulsar no, no hacemos nada
                SQLTables tables = new SQLTables(v.getContext(), "BD1", null, 3);
                SQLiteDatabase bdW = tables.getWritableDatabase();
            }
        });
    }
};
```

Ilustración 27: Fragmento de código 4

En este último caso, donde se produce la pulsación prolongada para eliminar una cita, además se puede observar cómo se utiliza la creación de un **AlertDialog**, o mensaje de confirmación que nos permitirá realizar el borrado en dos pasos, siempre que se cumpla o no una condición, y que hará que el usuario tenga la posibilidad de rectificar en su decisión de borrar una medida.

Otro aspecto reseñable dentro de nuestro código, será la creación de la base de datos que se encargará de almacenar todos nuestros datos registrados en el sistema. Esta clase implementará las diferentes tablas y las creará siguiendo la sintaxis SQL más tradicional, y hará que de una manera muy simplificada tengamos acceso a la misma, mediante la implementación de instancias de lectura y de escritura sobre la misma, como puede observarse en la imagen anterior. A continuación, se muestra la clase en la que se crea nuestra base de datos, con las diferentes tablas que hemos implementado para desarrollar el sistema:

```
public class SQLTables extends SQLiteOpenHelper{

    String sqlCreateUsuario = "CREATE TABLE Usuario ( nombre VARCHAR(50) NOT NULL, apellido VARCHAR(50) NOT NULL, " +
        "edad INTEGER NOT NULL, unidades INTEGER NOT NULL, " +
        "CONSTRAINT PK_USUARIO PRIMARY KEY (nombre,apellido,edad))";

    String sqlCreateMedidas = "CREATE TABLE Medidas (glucemia INTEGER NOT NULL, " +
        "fechaM DATE NOT NULL, horaM VARCHAR(20) NOT NULL, descripM VARCHAR(500)," +
        "CONSTRAINT PK_MEDIDAS PRIMARY KEY (glucemia,fechaM,horaM))";

    String sqlCreateCitas = " CREATE TABLE Citas (doctor VARCHAR(50) NOT NULL, " +
        "hospital VARCHAR(50) NOT NULL,fechaC DATE NOT NULL, " +
        "horaC TIME NOT NULL, descripC VARCHAR(500))," +
        "CONSTRAINT PK_MEDIDAS PRIMARY KEY (doctor, hospital, fechaC, horaC))";

    public SQLTables(Context context, String name, CursorFactory factory, int version) {
        super(context, name, factory, version);
    }

    public SQLTables(Context context, String name, CursorFactory factory, int version, DatabaseErrorHandler errorHandler) {
        super(context, name, factory, version, errorHandler);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        //Se ejecuta la sentencia SQL de creación de la tabla
        db.execSQL(sqlCreateUsuario);
        db.execSQL(sqlCreateCitas);
        db.execSQL(sqlCreateMedidas);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        //Se elimina la versión anterior de la tabla. Se borrará todo
        db.execSQL("DROP TABLE IF EXISTS Usuario");
        db.execSQL("DROP TABLE IF EXISTS Citas");
        db.execSQL("DROP TABLE IF EXISTS Medidas");

        //Se crea la nueva versión de la tabla. Se actualizará el String de creación
        db.execSQL(sqlCreateUsuario);
        db.execSQL(sqlCreateCitas);
        db.execSQL(sqlCreateMedidas);
    }
}
```

Ilustración 28: Fragmento de código 5

En este caso, se puede observar cómo se crea la base de datos en el momento en el que se instancia, pero además la capacidad para actualizar la base de datos con las diferentes versiones que podamos requerir en caso de realizar cambios sobre la estructura y diseño de la misma. En dicho caso, la metodología seguida será eliminar por completo la base de datos e insertar la actualización de las tablas de la misma, pero teniendo un control de versiones para saber cuándo se han modificado estas.

Pero el sistema de programación Android, al combinar la interfaz gráfica y la interfaz funcional, utiliza para dicho propósito la clase **R**, permitiendo las interacciones entre

ambas implementaciones. De esta manera, y en cada uno de las actividades que compondrán nuestro programa, se ejecutará como primera secuencia de código la mostrada en la siguiente imagen, que creará este vínculo, y permitirá la interacción entre la funcionalidad y la interfaz.

```
public class Graficos extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_graficos);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbarGraficos);
        setSupportActionBar(toolbar);
    }
}
```

Ilustración 29: Fragmento de código 6

Por último, es importante destacar uno de los aspectos más distintivos de nuestro programa, como es la capacidad de reproducir de manera gráfica los datos que se han registrado durante el tiempo por parte de los usuarios, de manera que se creen gráficas sobre los mismos. Este proceso puede parecer simple, pero tiene una complejidad mayúscula en cuanto a términos de programación se refieren. Para ello se ha recurrido a una librería externa capaz de simplificar el proceso de manera exponencial. Dicha librería ha sido descargada como código libre desde la página web GitHub, y su nombre es **MPAndroidChart** [35], siendo **Phil Jay** su autor. Para observar las grandes capacidades de esta librería se invita a visitar la página web y el repositorio de susodicho desarrollador, pero en este caso se proporciona a continuación como se han implementado algunos aspectos de las gráficas para nuestro sistema.

```
LineDataSet dataSet = new LineDataSet(representacion, "Medidas");
LineData lineData = new LineData(dataSet);
grafica.setData(lineData);
dataSet.setValueTextColor(Color.BLACK);
grafica.setGridBackgroundColor(Color.BLACK);
dataSet.setColor(Color.BLACK);
dataSet.setDrawFilled(true);
dataSet.setValueTextSize(15f);
dataSet.setLineWidth(2f);
//grafica.setAutoScaleMinMaxEnabled(true);

//Eje X
XAxis xAxis = grafica.getXAxis();
xAxis.setEnabled(true);
grafica.getXAxis().setPosition(XAxis.XAxisPosition.BOTTOM);
grafica.setVisibleXRange(0, 10);
xAxis.setGranularityEnabled(true);
xAxis.setGranularity(1f);
xAxis.setLabelCount(10, true);
}
```

Ilustración 30: Fragmento de código 7

```
entries.add(new PieEntry(media70, "<70"));
entries.add(new PieEntry(porcent70120, "70-120"));
entries.add(new PieEntry(porcent120170, "120-170"));
entries.add(new PieEntry(porcent170250, "170-250"));
entries.add(new PieEntry(porcent250, ">250"));

}

PieDataSet dataset = new PieDataSet(entries, "");

PieData data = new PieData(dataset);

Description desc = new Description();
desc.setText("Distribucion porcentual de medidas");
desc.setTextSize(15f);
desc.setTextColor(Color.BLACK);
pieChart.setDescription(desc);

pieChart.getLegend().setEnabled(false);

pieChart.setData(data);
pieChart.animateY(5000);
dataset.setColors(ColorTemplate.COLORFUL_COLORS);
dataset.setValueTextSize(20f);

}

}else{
    pieChart.setVisibility(View.GONE);
    grafica.setVisibility(View.VISIBLE);
}
```

En estas pantallas puede observarse la alta capacidad de modificación de muchísimos aspectos para la representación de los gráficos, tales como el tipo de gráfica, el fondo de nuestro diseño o la distancia entre las líneas de división, los rangos de datos, el nivel de precisión del zoom, o la diferencia máxima entre valores de representación en la gráfica, o el trabajo independiente con los ejes, aplicando valores únicos a cada

uno de ellos. Estos son solo algunos ejemplos de las ilimitadas posibilidades de esta librería tan completa, y tan utilizada por la comunidad de desarrollo en este lenguaje. Cabe referir que las opciones varían en función del tipo de gráfica que se utilice, siendo en nuestro caso la imagen izquierda una gráfica de tipo lineal, y en la imagen derecha una gráfica de pastel o circular.

5- Pruebas del sistema

El desarrollo de todo nuestro sistema software ha analizado y definido un planteamiento de solución para la implementación del sistema pretendido, definiendo un conjunto de requisitos, y casos de uso en la primera parte, y una definición arquitectónica, combinando los primeros entre sí. Pero todo este conjunto para poder definirse como solución válida, primero ha de ser verificado por un conjunto de pruebas capaces de determinar si la solución es viable o no.

Durante este apartado se va a definir dicho conjunto de pruebas, capaces de ofrecer una demostración sobre la completitud e integridad del sistema, basándonos en la verificación y validación de los requisitos diseñados durante el análisis del sistema.

Esta verificación y validación se centrará en dos partes diferenciadas, la primera se centrará en definir un conjunto de casos de prueba, a través de un conjunto de tablas. De esta manera y junto con la explicación sobre cada uno de los casos de prueba, pueden definirse un conjunto amplio de comprobaciones, capaz de registrar los diferentes casos de uso y requisitos de la aplicación, en los que se prueba su correcto funcionamiento.

Como segunda parte de nuestro apartado de pruebas, definiremos el conjunto de matrices de trazabilidad entre los requisitos y los casos de prueba, así como entre los casos de uso y los casos de prueba. Esto nos permitirá conocer a modo de resumen como se relacionan los requisitos, los casos de uso y los casos de prueba, verificando cada uno de ellos y obteniendo como resultado la comprobación de que el sistema es consistente, coherente y trazable.

5.1-Definición de casos de prueba

Caso de prueba 1: Este caso de prueba se centra en comprobar una gran cantidad de requisitos, forzando al sistema a ser iniciado por primera vez y tratando de crear un usuario. En este caso, se fuerza a la aparición del tutorial de uso para los usuarios primerizos, y forzándoles a realizar el primer registro en el sistema de sus datos para quedar registrados. Además de esto, el sistema nos notificará que se va a utilizar el almacenamiento interno para almacenar información, hecho para el cual tendremos que dar nuestra aprobación como usuarios.

Número de identificación de caso de prueba	CP-01
Descripción de caso de prueba	El caso de prueba consistirá en el inicio del sistema y registro de un usuario en el menú destinado para ello
Requisitos y casos de uso que cubre	Requisito 14 Requisito 16 Requisito 12 Caso de uso 1
Valor esperado	El valor esperado en esta ejecución se centra en que el usuario quede registrado en la base de

	datos, reflejándose esto en los campos de la aplicación, así como la no necesidad de volver a reproducir el tutorial de inicio, ni a aceptar la utilización de la memoria interna por parte de la aplicación
Valor obtenido	Se obtienen los resultados esperados, teniendo un usuario registrado en el sistema y mostrando esto en la aplicación, así como la evitación de mostrar el tutorial una vez se ha superado el primer inicio

Tabla 42: Caso de prueba 1

Caso de prueba 2: Este caso de prueba se centra en verificar la inserción de las citas en el sistema, verificando de esta manera la inserción en la base de datos, así como la posibilidad de visualización de estos registros en los espacios destinados para ello. Además de esto trataremos de mostrar los datos de la descripción de la cita almacenada para comprobar que se ha guardado y el sistema funciona correctamente y que los datos se pueden eliminar mediante la pulsación prolongada.

Número de identificación de caso de prueba	CP-02
Descripción de caso de prueba	El caso de prueba consistirá en la inserción de una cita en el sistema, siguiendo los pasos definidos para ello, y su posterior eliminación
Requisitos y casos de uso que cubre	Requisito 17 Requisito 19 Requisito 20 Requisito 21 Caso de uso 2 Caso de uso 4 Caso de uso 5
Valor esperado	El valor esperado en esta ejecución se centra en que la cita quede registrada en la base de datos y en el sistema, en forma de tarjeta y en el espacio reservado para ello
Valor obtenido	Se obtienen los resultados esperados, obteniendo una tarjeta nueva con los datos introducidos, característicos de la cita, y siendo borrados después

Tabla 43: Caso de prueba 2

Caso de prueba 3: Este caso de prueba se centra en verificar la inserción de las medidas en el sistema, verificando de esta manera la inserción en la base de datos, así como la posibilidad de visualización de estos registros en los espacios destinados para ello. Además de esto trataremos de mostrar las notas de la medida almacenada para comprobar que se ha guardado y el sistema funciona correctamente, y la posterior eliminación del registro.

Número de identificación de caso de prueba	CP-03
Descripción de caso de prueba	El caso de prueba consistirá en la inserción de una medida en el sistema, siguiendo los pasos definidos para ello, y su posterior eliminación
Requisitos y casos de uso que cubre	Requisito 18 Requisito 19 Requisito 20 Requisito 21 Caso de uso 3 Caso de uso 4 Caso de uso 5
Valor esperado	El valor esperado en esta ejecución se centra en que el registro glucémico quede reflejado en forma de tarjeta, para poder ser visualizado en el espacio y menú destinado para ello
Valor obtenido	Se obtienen los resultados esperados, teniendo una nueva tarjeta que mostrará los datos del registro introducidos por el usuario, siendo borrados después

Tabla 44: Caso de prueba 3

Caso de prueba 4: Este caso de prueba se centra en verificar la creación de las gráficas personalizadas a través de los datos introducidos en el sistema por parte del usuario. Para ello se introducirán de manera progresiva un conjunto de datos, para verificar que el sistema los gestiona de manera correcta.

Número de identificación de caso de prueba	CP-04
Descripción de caso de prueba	El caso de prueba consistirá en la inserción progresiva de valores diferentes, y la comprobación en cada iteración de que se cumplen los valores esperados en las gráficas generadas
Requisitos y casos de uso que cubre	Requisito 18 Requisito 19 Requisito 20 Requisito 21 Caso de uso 6
Valor esperado	El valor esperado en esta ejecución se centra en que el conjunto de graficas generadas para las de tipo pastel y tipo lineal, sean correctos en las progresivas iteraciones

Valor obtenido	Se obtienen los resultados esperados, obteniendo para cada iteración los valores esperados por el usuario
-----------------------	---

Tabla 45: Caso de prueba 4

Caso de prueba 5: Este caso de prueba se centra en verificar que el sistema funciona en diferentes tipos de dispositivos, con variaciones en el tamaño de pantalla.

Número de identificación de caso de prueba	CP-05
Descripción de caso de prueba	El caso de prueba consistirá en la instalación de la aplicación en diferentes dispositivos, así como la comprobación mediante el simulador proporcionado por Android
Requisitos y casos de uso que cubre	Requisito 14 Requisito 15
Valor esperado	El valor esperado en esta ejecución se centra en que el sistema funcione correctamente en cada uno de los dispositivos en los que se ha probado
Valor obtenido	Se obtienen los resultados esperados, obteniendo una ejecución similar en cada uno de los dispositivos y simulaciones

Tabla 46: Caso de prueba 5

Caso de prueba 6: Este caso de prueba se centra en verificar que el sistema rinde de manera correcta y eficiente en los dispositivos, valorando aspectos como el consumo de memoria RAM y el tiempo de interacción con la base de datos destinada a almacenar los datos del usuario.

Número de identificación de caso de prueba	CP-06
Descripción de caso de prueba	El caso de prueba consistirá en la medición de varios valores, tales como el consumo de memoria RAM en el sistema, proporcionado por el simulador Android Studio, y la medición de los tiempos de interacción con la base de datos
Requisitos y casos de uso que cubre	Requisito 8 Requisito 9
Valor esperado	El valor esperado en esta ejecución se centra en que el sistema funcione correctamente, con valores no críticos en cuanto a niveles de consumo de memoria RAM, e interacciones menores a tres segundos con la base de datos
Valor obtenido	Se obtienen los resultados esperados tras varias pruebas de medición y comprobación de consumo

Tabla 47: Caso de prueba 6

Caso de prueba 7: Este caso de prueba se centra en verificar que el sistema rinde de manera correcta y eficiente en el caso de querer realizar una copia de seguridad y restaurarla después del sistema en la nube.

Número de identificación de caso de prueba	CP-07
Descripción de caso de prueba	El caso de prueba consistirá en recuperación de los datos del usuario tras la inserción de un registro posterior a dicho backup
Requisitos y casos de uso que cubre	Requisito 11 Requisito 24 Requisito 25 Caso de uso 7
Valor esperado	El valor esperado en esta ejecución se centra en que el sistema recupere de forma correcta los últimos valores almacenados en la nube como copia de seguridad
Valor obtenido	Se obtienen los resultados esperados, obteniendo los valores disponibles en la última copia de seguridad

Tabla 48: Caso de prueba 7

Tras esta definición de los diferentes casos de prueba que se han realizado, cabe destacar que se han introducido secciones de código que acceden a la base de datos, durante las diferentes ejecuciones, para comprobar que los registros de esta han sido actualizados en el momento en el que se realizan operaciones de inserción o eliminación de datos.

Caso de prueba 8: Este caso de prueba se centra en verificar que el sistema ofrece información en el momento de pulsar sobre los botones de ayuda o información situados en los menús de nuestro sistema

Número de identificación de caso de prueba	CP-08
Descripción de caso de prueba	El caso de prueba consistirá en mostrar la información sobre aspectos que puedan afectar a la salud del paciente
Requisitos y casos de uso que cubre	Requisito 13
Valor esperado	El valor esperado en esta ejecución se centra en que el sistema muestre información sobre el botón de ayuda al que referencia
Valor obtenido	Se obtienen los resultados esperados, obteniendo información sobre valores críticos o no entendibles por los usuarios

Tabla 49: Caso de prueba 8

5.2-Trazabilidad del desarrollo del sistema

En la siguiente matriz se visualiza la trazabilidad entre los requisitos y los casos de uso. En ella cabe destacar el hecho de haber agrupado los requisitos específicos del sistema, ya que son cumplidos por todos los casos de prueba al ser generales, además del requisito que guarda relación con los términos y condiciones, que de igual manera todos los casos de prueba respetan.

5.2.1-Matriz de trazabilidad: Requisitos/Casos de prueba

En la siguiente matriz se ofrece la trazabilidad a nivel de requisitos del sistema, mostrando que casos de prueba han verificado y comprobado la solidez del mismo.

	CP-01	CP-02	CP-03	CP-04	CP-05	CP-06	CP-07	CP-08	CP-09
Requisitos 1-7	✓	✓	✓	✓	✓	✓	✓	✓	✓
Requisito 8		✓	✓						
Requisito 9						✓			
Requisito 10						✓			
Requisito 11							✓		
Requisito 12	✓								
Requisito 13								✓	
Requisito 14	✓								
Requisito 15					✓				
Requisito 16	✓								
Requisito 17		✓							
Requisito 18			✓						
Requisito 19		✓	✓						
Requisito 20		✓	✓						
Requisito 21		✓	✓						
Requisito 22				✓					
Requisito 23				✓					
Requisito 24							✓		
Requisito 25							✓		
Requisito 26	✓	✓	✓	✓	✓	✓	✓	✓	✓

Tabla 50: Matriz de trazabilidad Requisitos-Casos de prueba

5.2.2-Matriz de trazabilidad: Casos de uso/Casos de prueba

En esta matriz se ofrece la trazabilidad de los casos de uso, mostrando que pruebas se han debido realizar para comprobar y verificar la solidez del sistema, obteniendo resultados positivos en sus resultados.

	CP-01	CP-02	CP-03	CP-04	CP-05	CP-06	CP-07	CP-08
CU-01	✓							
CU-02		✓						
CU-03			✓					
CU-04		✓	✓					
CU-05		✓	✓					
CU-06				✓				
CU-07							✓	

Tabla 51: Matriz de trazabilidad Casos de uso-Requisitos

6- Marco legal del sistema

El desarrollo de cualquier sistema software, tiene un conjunto de requerimientos e interacciones con el usuario que han de desarrollarse para poder ser funcionales y poder cubrir una necesidad para el usuario. Pero estas interacciones y funcionalidades, han de establecerse y generarse siguiendo un conjunto controlado de normas y regulaciones para respetar varios aspectos que involucran al usuario y se amoldan a las normas establecidas para la vida en sociedad.

Uno de los aspectos más reseñables de este apartado se centra en los derechos sobre los datos de los usuarios, los derechos de propiedad intelectual, y la definición clara y concisa de un conjunto de términos legales y de condiciones, capaz de hacer saber al usuario que clase de restricciones y usos pueden darse a la aplicación, así como qué conjunto de datos y su uso va a requerir la aplicación, permitiendo al usuario final dar el visto bueno para dicha utilización.

De esta manera el primero de los aspectos a tratar es el establecer un conjunto de derechos sobre los datos del usuario. Este conjunto de aspectos y términos, que puede estar recogido en el documento referido al dicho propósito, se centrará en permitir al usuario conocer el conjunto de datos que van a ser utilizados por parte del sistema. En nuestro caso se almacenarán los nombres, apellidos, edad y datos médicos del usuario, siendo estos utilizados para la creación de estimaciones sobre las necesidades que puede tener el paciente en ciertos aspectos de la enfermedad que padece, y poder ofrecerle como se ha comentado a lo largo del documento, datos y valoraciones de mayor calidad. Este conjunto de datos únicamente será compartido en nuestro caso con el sistema de copias de seguridad que la aplicación proporciona, gestionado de manera segura, y nunca con otro sistema o usuario. Este desarrollo nos ayudará a fomentar la confianza que el cliente usuario deposite en nuestro sistema, pues garantizamos el uso no fraudulento de los datos por parte de cualquier otro sistema, pues el hecho de tratar con datos de carácter clínico, cuya sensibilidad es crítica, hacen aún si cabe más importante este aspecto.

Otro de los aspectos más importantes es la definición y registro de los derechos de propiedad intelectual de nuestro sistema. Con este conjunto de normas, deben registrarse todos los aspectos que caractericen y no hayan sido obtenidos de terceros en nuestra aplicación, tales como su nombre, logo, etc. En definitiva, el conjunto distintivo de aspectos que harán identificable a la aplicación ante las demás, protegiendo nuestro contenido. Esto se debe desarrollar con el propósito de protegerla ante posibles plagios que puedan producirse a diferentes niveles, y diferenciando los derechos propios ante nuestra creación. De igual manera, en caso de hacer uso en nuestra aplicación de contenidos de otro creador, sean del tipo que sean, debemos referenciarlo correctamente, poseer los permisos y licencias correspondientes.

En el anterior apartado, también podría implementarse un registro de patentes, permitiendo a la aplicación ser registrada como una idea no patentada, y haciendo que su uso fraudulento o su utilización por parte de otros usuarios reporte algún tipo de beneficio al ser una idea registrada. En nuestro caso este aspecto no se va a tratar, pues la idea del sistema es su implementación como un software accesible y utilizable de manera libre por otros desarrolladores para poder potenciar el desarrollo y mejora del sistema, haciéndolo útil quizás para un mayor número de pacientes.

El siguiente aspecto que podríamos tratar en este apartado se centra en la clasificación de nuestra aplicación en sistema proporcionado por IARC (International Age Rating Coalition) cuyo propósito es el de categorizar y administrar la aplicación, en diferentes niveles de utilización en relación con la edad. Esta organización permitirá obtener la certificación que dictará que usuarios son aptos para utilizar la aplicación, de tal manera que el uso de la misma quede restringido a los usuarios que cumplan con dicha condición, y en ningún caso puedan aplicarse acciones legales por desinformación hacia el usuario.

Un aspecto que debemos desarrollar para el uso de nuestra aplicación, se centra en definir un conjunto de términos y condiciones, cuyo objetivo se centra en registrar el conjunto de normas de utilización y necesidad que tiene nuestra aplicación y que, en caso de ser aceptadas por el usuario, permitirán al desarrollador o empresa abstenerse de cualquier acción legal emprendida por el usuario en caso de estar reflejada en dicho conjunto. Este es uno de los aspectos más destacables de este apartado, pues en el deben registrarse el conjunto general de normas del sistema, necesidades sobre los datos de usuario y la protección de los mismos, los vínculos externos utilizados por la aplicación, etc.

En nuestro caso esta definición cobra una importancia aún más grande pues estamos usando datos de carácter clínico, por lo que el usuario debe prestar mucha atención antes de dar su visto bueno a la utilización. El apartado de datos se puede encontrar en el documento de derechos sobre los datos del usuario. Con respecto a los vínculos externos que utilizaremos en nuestro sistema, cabe destacar que nuestra aplicación ha sido diseñada para el uso en un único dispositivo, evitando la interconexión de dispositivos, para de esta manera evitar este conjunto de posibles ataques a la privacidad del usuario que puedan producirse. Existe además un conjunto de requisitos de seguridad presentes en este documento, que garantizarán, dado su cumplimiento, que este uso fraudulento no se produzca.

Por último y como aspecto totalmente necesario en este documento, se dicta que nuestra aplicación en ningún momento ha sido diseñada para utilizarse como aplicación única a la hora de realizar recomendaciones sobre la salud de los pacientes. Es por esto que todo el conjunto de datos que se refleje en la aplicación, debe ser totalmente contrastado y valorado antes de la aplicación de cualquier medida correctora hacia la enfermedad, por parte del usuario.

Este sistema únicamente servirá como ayuda y apoyo para el enfermo, y en ningún caso sustitutivo a su conocimiento y experiencia sobre la enfermedad, por lo que el juicio del paciente tiene un valor mayor a la aplicación, destacando que esta aplicación pretende informar al usuario para que pueda tomar mejores decisiones para el control de la enfermedad. De igual manera, se recomienda al paciente establecer un criterio médico real ante los valores obtenidos y controlados por dicha aplicación, de tal manera que los datos obtenidos puedan contrastarse por un profesional.

De esta manera, además, cabe resaltar que el desarrollador no se hace responsable de la pérdida o extravío del dispositivo físico, o los ataques maliciosos que se lancen contra el dispositivo por otros medios diferentes a los del propio sistema, por lo que, en tales casos, la protección de datos de este deja de tener efectos prácticos.

Por último, destacar que el conjunto citado de términos y condiciones de uso puede ser modificado en cualquier momento por medio del desarrollador, transmitiendo esto a los usuarios para que estos puedan aceptarlos una vez modificados, sabiendo que cualquiera norma anteriormente escrita deja de tener valor, y únicamente se tendrá en cuenta la última versión generada y aceptada.

A continuación, se presentan los artículos y leyes de obligado cumplimiento por la aplicación:

- Ley 41/2002, de 14 de noviembre, básica reguladora de la autonomía del paciente y de derechos y obligaciones en materia de información y documentación clínica. [36]
- Directiva 2002/58/CE del Parlamento Europeo y del Consejo, de 12 de julio de 2002, relativa al tratamiento de los datos personales y a la protección de la intimidad en el sector de las comunicaciones electrónicas (Directiva sobre la privacidad y las comunicaciones electrónicas). [37]
- Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal. [38]
- Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data. [39]

7- Entorno socio económico

El desarrollo de sistemas o proyectos software en la actualidad, viene directamente ligado a un conjunto de mejoras, funcionalidades, o aplicaciones de una tecnología sobre las necesidades humanas. El conjunto de hábitos sobre el que desarrollamos nuestra vida, nos permite desarrollar una serie de tareas y satisfacer las mismas. Pero los proyectos software, a su vez, están vinculados directamente a un intangible que no se ha valorado antes en ninguno de los anteriores apartados: Las capacidades vinculadas al entorno en el que se dispondrán y su influencia en la economía de mercado en la que se centraliza su uso. Ambos aspectos a su vez están relacionados entre sí, pues dependen el uno del otro, por lo que la mejora de los dos en conjunción, mostrarán unos resultados mucho más concluyentes. De esta manera, durante este apartado del documento, vamos a observar como nuestro sistema puede generar soluciones ante diferentes problemas o posibles entornos de mejora con los que está directamente relacionado.

Durante el último siglo y debido a los grandes avances tecnológicos, la salud humana se ha visto directamente repercutida, dadas las facilidades y comodidades que han generado en la sociedad los mismos. El hecho de utilizar el coche o el autobús, en lugar de caminar, o realizar compras a través de la web en lugar de ir al supermercado, son hábitos en el actual nivel de vida de la sociedad media, que influyen directamente en el sedentarismo progresivo que estamos sufriendo. Este sedentarismo a su vez está potenciando un aumento de enfermedades relacionadas directamente con el sobrepeso u obesidad, entre las cuales se incluye la diabetes [40]. Pero además este problema se ve claramente relacionado con el nivel de desarrollo de los países, pues existe un índice mucho más alto de población que padece dicho tipo de enfermedades. Otra de las causas principales también vinculadas con este tipo de enfermedades, se basa en la cada vez peores hábitos alimenticios que nuestra sociedad está tomando, a un ritmo potencialmente peligroso. Los restaurantes de comida rápida, los alimentos excesivamente procesados, las industrias alimenticias, etc. están creando una sociedad en riesgo que sin ninguna duda debe frenarse cuanto antes.

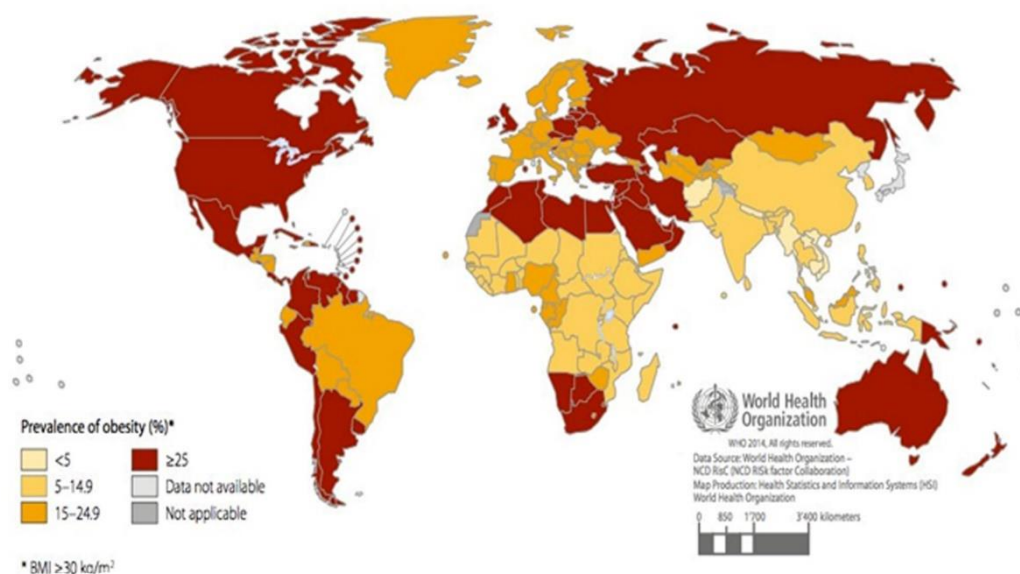


Ilustración 31: Los problemas de la obesidad a nivel mundial según la OMS [44]

Es por tanto que este conjunto de malos hábitos presentes en nuestra sociedad, generando un gran problema de obesidad y sobrepeso, han aumentado en gran medida el número de pacientes diabéticos. La organización mundial de la salud (**OMS**) calcula que en torno al **39%** de la población mundial mayor de 18 años, padecía sobrepeso, de entre las cuales un **13%**, sobrepasaban los límites de la obesidad, y que entorno al **8,5%** de la población es diabética, lo que se traduce en un total de 422 millones de pacientes [6]. Ambos datos están totalmente ligados a estadísticas que no paran de crecer.

Esta situación, por tanto, debería ser especialmente preocupante, debido a que se sabe que la diabetes y sus problemas derivados, son causantes de alrededor de **3** millones de muertes durante el año 2012, cifra potencialmente más elevada dentro del ritmo actual de crecimiento de la enfermedad.

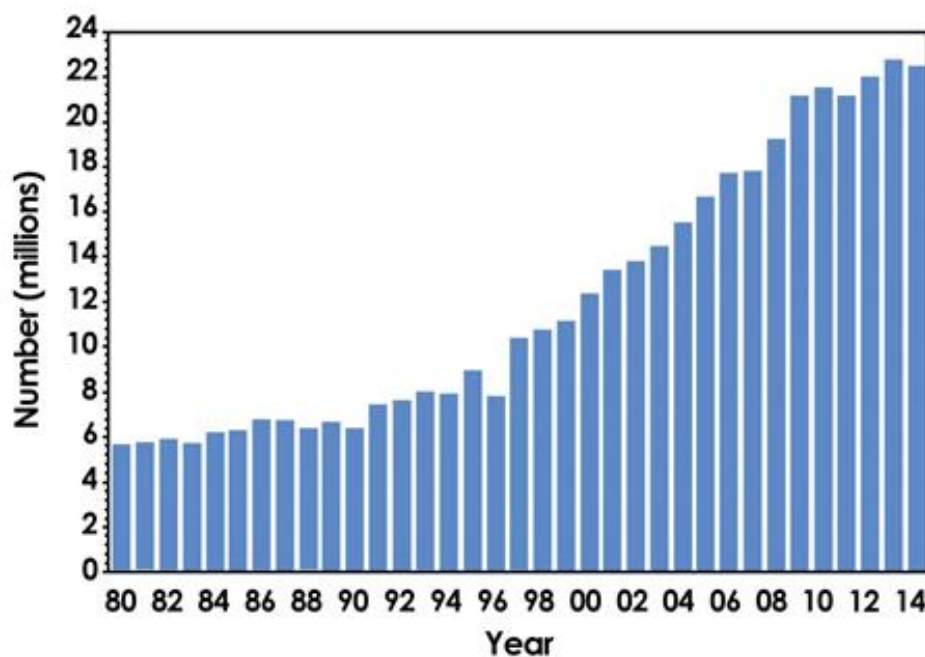


Ilustración 32: El aumento progresivo de la diabetes como enfermedad [43]

De esta manera, nuestro sistema será capaz de proporcionar un nuevo método de registro y control de datos para este gran conjunto de pacientes, descendiendo la criticidad de la aplicación a nivel mundial, y haciendo que el impacto en nuestra sociedad pueda ser lo menor posible.

Pero existe otro aspecto que cobra una importancia mayúscula en el desarrollo de una aplicación de estas características, este aspecto se centra en el concepto de **salud digital** [41]. Esta iniciativa surge a través de los conceptos de vincular los grandes niveles de digitalización y desarrollo tecnológico, con los servicios sanitarios que, como se explica, cada vez son más necesarios en la sociedad actual, no solo por la prevención y tratamiento de enfermedades crónicas, si no por factores como el envejecimiento y necesidades clínicas de la sociedad, el saneamiento de costes y privatización del sector, etc. Fruto de este conjunto de factores, nace dicha iniciativa, que pretende establecer una comunidad global, lo más amplia posible, dedicada concretamente a este sector tan necesario y demandado. Cabe resaltar que la iniciativa a nacido a través de multitud de profesionales del sector, favoreciendo a su vez el crecimiento de la investigación en todos los ámbitos, generando nuevas posibilidades de control, avance, información e innovación.

Esta iniciativa es uno de los puntos de interés para la creación de nuestro sistema, que pretende por encima de cualquier otra necesidad, favorecer el crecimiento y concienciación sobre la enfermedad que las personas diabéticas padecen, por tanto, es sin duda un gran aporte en cuanto a su impacto en la sociedad, ya que el proyecto puede apoyarse en una comunidad profesional inmensa tras él.

De esta manera el concepto anteriormente citado, pondría de manifiesto la facilidad para una necesidad que toda paciente demanda, como es la sanidad pública y el

derecho a poseer a través de impuestos a unas necesidades clínicas óptimas. Este concepto implantado en gran cantidad de países, no es tan común en países de bajo desarrollo o producto interior bruto bajo, por lo que, en dichos países, la sociedad es más vulnerable sobre el padecimiento de enfermedades de esta índole. Por este motivo nuestro sistema pretende entrar de lleno y ofrecer un nuevo conjunto de posibilidades para los pacientes, limitando el número de visitas o revisiones que por ejemplo, una persona pueda necesitar realizar, para el correcto control de su enfermedad, o a través de la intercomunicación con médicos o centros especializados, favoreces el conocimiento de que grupos sociales o tipos de pacientes pueden ser más susceptibles de necesitar dichos servicios, ahorrando de esta manera una gran cantidad de costes, y ofreciendo para los pacientes que realmente lo necesitan, un mejor servicio.

Pero el avance tecnológico no solo podría ayudar a los pacientes y a los gobiernos a ahorrar costes en términos de sanidad pública, sino que además abriría un nuevo campo a la distribución y gestión de los medicamentos y accesorios necesarios para el control de este tipo de enfermedades. Para una persona diabética, el porcentaje más alto de costes para el correcto mantenimiento de su salud viene derivado de la necesidad de materiales clínicos, tales como tiras reactivas, aparatos de medición, o la insulina inyectable.

The list price of Humalog insulin keeps going up

Since 1996, there have been more than two dozen price increases on a vial of Humalog insulin. Adjusted for inflation, the current price is 700% higher than it was 20 years ago.

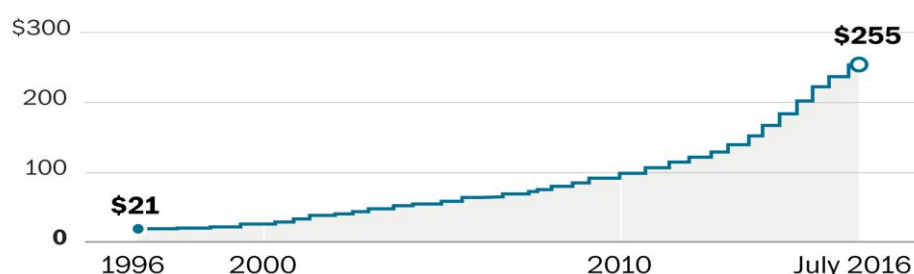


Ilustración 33: Aumento del coste de la insulina Humalog [42]

Este material no se caracteriza por su bajo coste, estando en torno a los 20-25 euros una simple pluma de insulina, por lo que la distribución y su regulación puede ser crítica para muchos pacientes.

Así pues, tras analizar este entorno en el que se desenvuelve nuestro sistema diseñado, debemos añadir que las características que ofrece nuestra aplicación son únicamente un primer paso para la generación de una idea o proyecto común, el poder establecer una comunidad de desarrollo sobre el mundo de la diabetes. Por tanto, cabe resaltar que nuestro producto será tratado como un software de código libre, re utilizable y modificable a través de los medios electrónicos y el desarrollo personal, y que tratará de evolucionar y desarrollarse, en el mayor grado posible, con

el apoyo de la comunidad sanitaria, complementándose, y aplicando nuevas soluciones ante problemas que puedan surgir durante la evolución de esta enfermedad. De esta manera el enriquecimiento mutuo para hacer avanzar un proyecto capaz de englobar a un conjunto tan grande de personas, es totalmente necesario, y el lucro a través de un sistema que pueda salvar la vida diariamente de cientos de personas, no tiene espacio en este proyecto.

8- Presupuesto y planificación

8.1-Presupuesto

Durante el siguiente apartado, se realizará un desglose del presupuesto empleado para el proyecto que se ha llevado a cabo, así como la visualización de un diagrama de Gantt, donde podrá observarse el proceso de desarrollo del mismo de manera temporal, en cada una de sus etapas.

Para la definición de este presupuesto, se definen varios conjuntos generadores de costes como son los costes de capital humano, en este caso del desarrollador, que en este caso es el propio alumno que ha implementado la aplicación y por otro lado el coste de capital no humano, en este caso material informático, como licencias y dispositivos utilizados para su desarrollo. De esta manera, obtendremos un desglose que tendrá la siguiente disposición:

8.1.1-Costes directos:

- Salario de desarrollador por hora de trabajo que será de 20€ por hora de trabajo realizado, por lo tanto, sabiendo que la media de horas de trabajo al día ha sido de 3 horas, tendremos:

$$213 \text{ días de desarrollo} * 3 \text{ horas/ día} * 20\text{€ /hora de trabajo} = \mathbf{12.780\text{€}}$$

- Para el desarrollo de nuestro sistema, se ha utilizado un ordenador portátil de las siguientes características:

Marca y modelos	Xiaomi Mi Notebook Air
Procesador	Intel Core i5-6200U (2.30GHz)
Memoria RAM	8 Gb GDDR4
Tarjeta gráfica	Nvidia GeForce 940MX (dedicada)
Tamaño de la pantalla	13.3 Full HD (1920x1080)
Memoria interna	256Gb SSD
Antigüedad	2017
Sistema Operativo	Windows 10 Home Premium 64 bits

Dado que el dispositivo ha sido comprado para el desarrollo del proyecto, se ha definido una parte del presupuesto a su compra, teniendo un valor de **624,50 €**, sobre el que se deberá calcular la amortización relativa al mismo, por tanto:

$$624,5 \text{ €} * 21\% / \text{año} * (213/360) \text{ días/año} = \\ 624,5 * 0,21 * 213/360 = \mathbf{77,59 \text{ €}}$$

- Licencia de Microsoft Office y Windows, adquiridas de manera **gratuita** gracias al convenio con la Universidad Carlos III de Madrid.

8.1.2-Costes indirectos:

- Coste de electricidad derivado de la utilización de las aulas donde se ha desarrollado el proyecto, que siendo un trabajo desarrollado completamente a nivel digital:

$$213 \text{ días} * 3 \text{ horas/ día} * 0,1\text{KW} * 0,12613 \text{ €/KWh} = \mathbf{8,05 \text{ €}}$$

- El coste del desplazamiento para las diferentes citas con el cliente, incluido en el coste humano, también se han añadido a este presupuesto, suponiendo un total de 74,2 Km por desplazamiento al destino de las reuniones, con un consumo medio de combustible de 6,5€ cada 100 Km, y un total de cuatro reuniones, lo que supondrá:

$$4 \text{ reuniones pactadas} * ((74,2\text{Km} * 6,5\text{€}) / 100 \text{ km}) = \mathbf{19,29\text{€}}$$

8.1.3- Beneficios

En nuestro caso este apartado, es un apartado muy abstracto de calcular, pues nuestro sistema nace con una intención no lucrativa. De esta manera para el cálculo del beneficio generado por nuestra aplicación, este se reducirá al saneamiento de los costes provenientes del desarrollo y la aplicación de los impuestos. Este modelo de negocio es seguido por muchas otras aplicaciones en su fase de crecimiento, tratando de atraer a un mayor número de usuarios hacia su utilización.

Pero en la práctica, pueden existir situaciones dada la aceptación de nuestro sistema, en la que pueda ser necesaria la aplicación de un valor de venta generador de beneficio. Este beneficio puede estar generado por un simple coste del producto, o por la utilización de compras in-app que puedan implementarse, permitiendo nuevas funciones. Este no es el objetivo principal con el que nace la aplicación, pero para este supuesto, vamos a realizar los cálculos de dicha ganancia posible suponiendo que el beneficio que queremos generar es de un **19%**.

$$(12.799,29+85,64) * 0,19 + 12.799,29+85,64 = \mathbf{15.333,07\text{€}}$$

Por tanto, obtendremos un beneficio de:

$$15.333,07 - 12857,59 = \mathbf{2475,48\text{€}}$$

De esta manera, a modo de resumen, nuestro coste de presupuesto será:

Origen del coste directo	Valor en euros
<i>Ordenador portátil</i>	77,59€
<i>Salario de los trabajadores</i>	12.780€
<i>Licencias</i>	0€

Tabla 52: Tabla de costes directos

Origen del coste indirecto	Valor en euros
Costes por desplazamiento	19,29€
Costes eléctricos	8,05€

Tabla 53: Tabla de costes indirectos

<u>Tabla de totales</u>	
<i>Total beneficios</i>	2.475,48€
<i>Total costes directos</i>	12.857,59€
<i>Total costes indirectos</i>	27,34€
<i>Total sin I.V.A</i>	15.360,41€
<i>Total con I.V.A</i>	18.586,09€

Tabla 54: Tabla de costes totales

Con respecto a la planificación del proyecto, a continuación, se dispone una tabla orientativa sobre las diferentes fases que se han seguido en el desarrollo del mismo, en la que se muestran un total de 213 días de desarrollo, combinando tareas de programación e implementación, y tareas de desarrollo del documento vinculado a dicho desarrollo. Se ha tenido en cuenta que la media de horas trabajadas durante estos periodos ha sido de aproximadamente 3 horas diarias, existiendo en ciertos días mayor cantidad de horas, y en otros menos.

Cabe destacar que, durante el desarrollo de este proyecto, ha existido un conjunto de procesos paralelos a la finalización del curso lectivo, por lo que han existido etapas en las que el tiempo de trabajo ha sido cero o prácticamente mínimo, pero aun así se han desarrollado tareas de corrección, estructuración o mejora del código, que se han incluido en la tarea ***Correcciones del código fuente***.

Como puede observarse en la siguiente tabla y su correspondiente diagrama Gantt, existe una clara división de tareas de programación y desarrollo de la memoria, siendo la primera etapa un desarrollo totalmente continuo, mientras que, en el segundo caso, realizando una subdivisión de tareas en diferentes periodos retroalimentados, para la generación del documento. Este proceso ha sido un proceso de desarrollo lineal, con corrección de errores en diferentes reuniones con el cliente, pero con un alto grado de dependencia entre las etapas de desarrollo.

Diagrama de Gantt			Duración (días)
Fecha de inicio	Fecha de Fin	Tarea	
19/11/2016	30/01/2017	Implementación código fuente	72
31/01/2017	20/06/2017	Correcciones código fuente	141
01/02/2017	07/02/2017	Primera reunión cliente	7
08/02/2017	14/02/2017	Introducción y objetivos del documento	7
17/02/2017	23/02/2017	Estado del arte	7
27/03/2017	10/04/2017	Análisis del sistema	15
11/04/2017	17/04/2017	Corrección de errores (I)	7
29/05/2017	10/06/2017	Diseño del sistema	13
08/06/2017	10/06/2017	Implementación	3
11/06/2017	13/06/2017	Pruebas	3
14/06/2017	16/06/2017	Definición del marco legal del sistema	3
14/06/2017	16/06/2017	Entorno socio-económico	3
17/06/2017	19/06/2017	Conclusiones	3
17/06/2016	19/06/2017	Resumen en inglés	3
19/06/2017	20/06/2017	Corrección de errores (II)	2

Tabla 55: Tabla Gantt

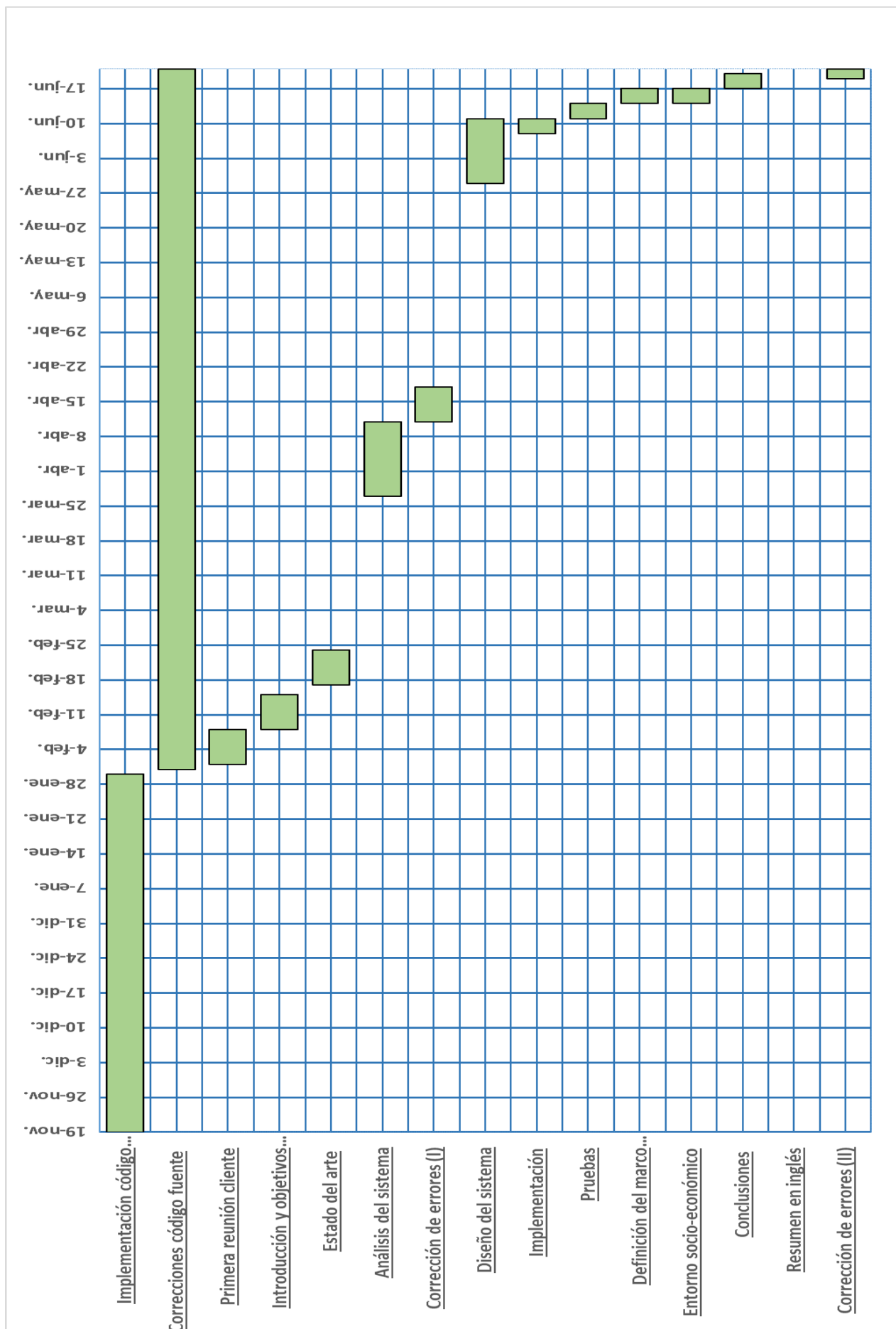


Ilustración 34: Diagrama de Gantt

9- Conclusiones y trabajo futuro

Durante todo este documento se ha expuesto la creación del sistema de registro y monitorización **“GDMSCT: GLUCOSE DATA MONITORING SYSTEM AND DIABETES CONTROL TIPS”** cuyo objetivo principal viene dado por la necesidad que existe actualmente en el sector hacia el que está enfocado. Dada la situación actual y el ritmo de vida hacia el que está enfocada la sociedad actual, siendo este cada vez más sedentaria, con una alimentación poco equilibrada y siendo víctima de los desequilibrios económicos, cada vez se demanda más la necesidad de implantar la tecnología en aspectos quizás menos extendidos de nuestras vidas, pues, al fin y al cabo, la tecnología tiene este único propósito, el de mejorar las vidas de las personas en las que está presente.

De esta manera la implementación y generación de esta idea permitirá a los pacientes que padezcan Diabetes Tipo 1, llevar un control glucémico monitorizado, pudiendo obtener mejores resultados en sus test, y en definitiva mejorando su nivel de vida a través de la salud.

Con respecto al desarrollo del proyecto, se ha tratado de definir un desarrollo adecuado para el tipo de público al que está enfocado, y dando la posibilidad al mayor número de personas para utilizar el sistema. Por ello se decidió la implementación en Android, como sistema operativo más utilizado y modificable globalmente.

La mayor inversión de tiempo se utilizó en la creación de un diseño del sistema óptimo, consistente y sólido, que desarrollara el mayor número de funcionalidades necesarias posible desde el punto de vista del usuario final. Esta inversión de tiempo, ha supuesto un sistema adecuado y capaz, que ofrece al usuario otro punto de vista y abstracción hacia su enfermedad, y permite un mejor control de esta, al ofrecer datos relevantes y útiles para su corrección.

Los plazos que se plantearon en el proyecto han sido cumplidos, siendo un punto clave del desarrollo e implementación del mismo, pues la combinación con otros trabajos y proyectos de la universidad han estado presentes durante gran parte de dicho desarrollo.

Como último punto de este desarrollo, me gustaría realizar un llamamiento a la concienciación que debe tomar en primera instancia, la salud en nuestras vidas, pues gozando de ella las posibilidades de crecimiento en todos los demás aspectos que engloban a la sociedad estarían mucho mejor desarrollados. Pero en segunda estancia me gustaría realizar un llamamiento a la concienciación para el tratamiento de la Diabetes, cuya gravedad quizá no tenga la importancia en la mentalidad ciudadana, y actos tan simples como utilizar medios de transporte como la bicicleta, caminar o evitar los alimentos extra procesados, podrían evitar padecer dicha enfermedad.

Con respecto a las futuras aplicaciones que pueden derivar de la creación de este sistema, GDMSCCT pretende establecer una nueva visión, muy ligado al concepto de **Salud Digital**, de la que ya se ha hablado en este documento. La intención es aportar un primer paso a la creación de una comunidad de análisis, experimentación y concienciación, con la cual tratar la enfermedad de la Diabetes.

El hecho de que el sistema sea **Open Source** o de código libre, apoya al desarrollo, y permitirá en cualquier momento descargar de cualquier repositorio el código fuente, editarlo, modificarlo y lanzar nuevas versiones, mejoradas, o simplemente que ofrezcan nuevas capacidades y ventajas a la vida de los pacientes que deseen utilizarla. De esta manera, este software tendrá una capacidad prácticamente infinita para crecer y dar soporte a más usuarios, y haciéndolo de mejor manera.

Por otro lado, el sistema apoyará el análisis y la experimentación pues, el auge de las nuevas tecnologías de análisis masivo o **BigData**, podrían generar en un futuro una base de datos excepcional sobre datos captados por esta aplicación, pudiendo generar algoritmos de predicción o de **DeepLerning**, capaces de ayudar de una manera infinitamente más precisa a los pacientes.

Además de todo esto, el futuro de la aplicación es muy prometedor a nivel de usuario, pues este puede en un futuro, simplemente utilizar su dispositivo para registrar los datos, de una manera fácil y sencilla, evitando que empresas intermediarias deban someter a los pacientes a utilizar dispositivos propios, elevando el coste y lucrándose a su costa.

Introduction

During this project we will explain, the approaching with the programs that have led to the realization of this end grade project. Moreover, in some cases have qualified important aspects that our system can perform. It will show how the idea was born, the motivations, and some practical examples or similar applications that have been tested and tested order to have a rigorous and specific judgment of what was wanted to do.

The main objective is the development of a system capable of recording and monitoring in a useful way, basic aspects for the treatment of Diabetes Mellitus Type 1 or also known as juvenile. This system will allow us to register and manipulate the data stored in a much simpler way than usual in a desktop system, being possible its installation in mobile devices which will allow us to record measurements in real time.

Saying that there are some personal aspects that have influenced the realization of such a system. The fact that I personally suffer from this disease has helped and is served during my formative period. To realize that there is a lack of contributions to this particular field of science, or at least not entirely useful in certain aspects. My passion for portable devices, home automation or the rise of connectivity between devices of recent times, are just some examples that have made me wonder what could be the result in the case of combining or aligning these fields of knowledge, forming one unique and common one, that also can be helpful to the people who suffer from this disease.

In addition, there is a greater factor such as the possibility of the awareness of this disease in the current world, which is unfortunately not enough well-known even though it's suffer from a large number of people worldwide.

This eagerness to achieve a union between two areas very present in the life of how I say, large number of people, arises, as I say, the new technological trends that we have more present day in our lives. As we all know, life is constantly changing, and technology will be in charge of modernizing and facilitating more and more all the processes that the people have to carry out. And that is why adapting the health care environment should be a source of motivation for these advances, since it would improve the main and most important area of a person's life: the life. And all this with a portable device so that it will be easy and fast to use.

In this section, is where another motivation arises, which will explain another of the important points of the realization of our project. This is the use of a mobile operating system and not desktop. This aspect is quite frankly variable from one set to another of people, but most people worldwide are accompanied every day by their mobile phone or tablet, so the accessibility that this type of system contributes to the well-being of many people.

Finally, another of the most relevant facts that have led me to pretend to carry out such a system project, focuses on promoting "free software". While there is no doubt that a problem that is so common to many people, it cannot be dealt by only one so aid between members is vital for an application to be really useful becoming possible a feedback community between patients and developers.

Objectives

As explained in the previous section, the objective during this document will focus on the theoretical and practical definition of a recording and monitoring system for the control of a specific disease, Type 1 Diabetes Mellitus or juvenile.

Moreover, we analyzed the current state of the different factors discussed above, placing the reader on the subject that will be developed.

During the second part of the development, we will focus on the formal definition of our system, with special emphasis on what theoretical development is concerned, and trying to capture and explain all those relevant aspects, on which our software developed. This section will perform analysis, design, implementation and testing.

Lastly, in the last section of the document, the sections of development that deal with budgets, project planning, and the tasks performed during the development and implementation of our system will be available.

This will also include a section that will define the regulatory framework and the socioeconomic environment that has involved our system, an aspect of vital importance in such a system, since it deals with legally sensitive aspects that can cause privacy problems, as well as risks in the Health of persons in case of fraudulent use.

It should be noted that, during the development of the document, the parts in which we have invested the most time have focused on the analysis and design and implementation of the system.

During the first of these points, the objective has been to try to analyze the same, trying to obtain a defined set of requirements to ensure that it will be configured as a solid, stable and consistent system between its functionalities and within its software architecture.

On the other hand, during the development of the system and its implementation, we have tried to follow this previously performed software analysis, to generate a structured and well defined system with a consistent architecture.

Art State

Before making this project the first step was to investigate what the current situation was taking into account the different aspects of the disease, measurements tools and current hardware with its characteristics and uses. Finally, we will pay attention to the programming language in which the system has been developed, which is the Android language.

This disease is perhaps one of the most common in the today's world today, enhanced by the current way of life, but it is at the same time a known for the majority.

Diabetes mellitus -simply referred to as diabetes- is a physiological disorder focus on the lack of insulin production by the pancreas, or the inability to use properly the doses that this organ generates. The first case will be defined as Type I Diabetes Mellitus or juvenile, while the second type is treated as Type II Diabetes. Insulin is a type of hormone generated in this organ to allow the absorption of carbohydrates, as well as other substances necessary in each of the cells of our body to keep us alive, and to do so in a healthy way. This inconsistency between insulin levels and nutrients obtained from consumed foods can produce two types of processes in a person suffering from this disease, such as hypo and hyperglycemia. These focus on an excessive or insufficient level of insulin.

The first of them will make the body enter into a state of energy need, getting to produce feelings of weakness and fatigue, reaching in extreme cases the diabetic coma. Therefore, it is absolutely necessary the consumption of carbohydrates in such cases. On the contrary, the second case occurs in the opposite situation, where the insulin deficit will not allow the subject to absorb all of the nutrients, so that their blood cycle will carry an excess of carbohydrates, being equally dangerous, Although the symptoms are not so specific.

For both types of diabetes there is a set of risk factors that can promote the onset of this disease. Let's establish a brief summary of which are the most important:

For type I or juvenile diabetes, that is to say, in which the pancreas completely cancels the production of insulin, there are no clear factors about the origin in a user of the disease. But it is suspected and this is something that is still in a state of investigation, that the genetics or the antecedents of a relative who has suffered this disease may be the reason for the appearance of it. In addition, the possibilities on the exposure to different environmental factors or viral infections are valued can also potentiate the development of this variation of the disease.

According to IDF, the latest information says that approximately 415 million people suffer from the disease, whether type one or two, and this figure is gradually increasing at a very worrying rate.

This tendency, in turn, is associated with countries with a low GDP (Gross Domestic Product), which means that they are underdeveloped or developing countries, which have fewer means to help a larger population, this being in addition in continuous increase.

Another worrying fact is that one in two people who suffer from this disease are undiagnosed, that is, they do not know the symptoms and are unaware of the disease. This can cause serious problems in the health of people, to the point of death. This translates into five million people killed during 2015, or what is the same, a victim because of this disease every six seconds.

It is for this set of reasons why this disease has to be treated with other eyes and from another point of view, giving it the real importance it has, because of its seriousness and the group of people involved, of all ages and overall.

Analysis & Development

In this section the aim has been to develop a set of software requirements and use cases, which allows developers to get to know the set of specifications and characteristics of software developed in a concrete environment. So that it will result in a different software. They will help defining in a clear and concise way each of the products that have our software, making it respect, and being coherent, precise, well designed and verifiable sea through specific checks. Finally, our set of specifications must be unique to show the place without doubt, all the structural elements that have our system, structuring the architecture, and being the basis of its development.

To begin with this section, our architecture will follow a methodology to be able to justify each of the treaties on the same. In this way, for the software definition of the requirements, we will follow the IEEE830-1998 standard, but in a simplified way and taking only some of the sections defined in it, eliminated the sections in which our system does not apply.

A list of 26 software requirements, including performance requirements, interface requirements, reliability, function requirements and constraint requirements, has been obtained by the respective analysis. In addition, there is a set that will be defined by the customer.

With regard to use cases, define a set of activities and functions of our software at the time of use. They show the different steps that lead the software to generate a response for the user.

After the analysis of the system, we may establish a set of steps to be able to implement its functionality, defining as already discussed, and well defined architecture generated through analysis. This process has followed, as in the case of the analysis, a clear methodology, the so-called 4 + 1 methodology which includes a definition of its elements, relationships and functions within the system, based on different points of view or levels of abstraction.

Therefore, this is a complete methodology that will allow a high precision, when treating the same development from different levels, studying each one of its points and making it clearer and better defined, allowing to create an architecture of higher quality and efficiency. The viewpoints that conform this methodology are the development view, in which the highest level functionalities of the system are observed. On the other hand, physical abstraction shows the interaction between different systems. Thirdly, reaching more concrete levels of abstraction, the logical view is defined, being represented by class and component diagrams, arriving at code level and, finally, the interaction between processes will be defined in the process view.

In addition, to define more clearly the operation of our system, we added a diagram called a screen map, which provides the screens available in our software system, and what are the possibilities of reaching each of them.

In this section a small summary of the capabilities of the technological environment in which we have developed our system has been defined. In our case this has been based on the use of Android Studio, a program developed by Google, own for the creation of applications in the Android operating system, and that includes a lot of advantages for this type of developments.

Results

The results of the development of our system are clear. The ability to obtain a broad set of requirements, covering all the possibilities and needs of the system, has allowed the generation of a solid architecture, reliable and with specific characteristics. This definition, therefore, has allowed us to generate software capable of providing the intended functions for the user.

The first one focuses on the generation of a glycemic measures registration system, with which the user is comfortable to make their additions in a simple way. This has been developed through a system of cards, defined in different ways according to the measure introduced by the user.

Another feature that has been developed is the ability to store doctor's appointments in the system, as a reminder, to make the user never forget them. This will be helpful mainly among elderly people because in young people this is not such a common problem.

Thirdly and with regard to the actual functionality of the application, it would be worth highlighting the definition of a section to monitor the data of the measures introduced by the user, generating graphs capable of showing a different approach for the patient, being able to visualize conflicting values or alterations in their levels.

Finally, it should be noted a functionality that refers to the security and reliability of the system, such as the ability to perform backups in the cloud, as well as the ability to store data in the system's own database. In this way, users can rest easy, without worrying about losing the records stored in the time, whose consequences could be a serious alteration in their levels, and consequent in their health.

We have reach this results after a long process of development and implementation, whose graphic representation can be shown in the Gantt diagram which, together with it, includes a breakdown of the budget, including total costs and benefits, as well as the application of taxes.

But one of the most important points in general is the legal section of the system. The development of any software system has a set of requirements and interactions with the user that have to be developed in order to be functional and able to cover a need for the user. But these interactions and functionalities have to be established and generated following a controlled set of rules and regulations to respect various aspects that involve the user and conform to the norms established for life in society.

One of the most noteworthy aspects of this section is the rights to user data, intellectual property rights, and the clear and concise definition of a set of legal terms and conditions, capable of letting the user know that Class of restrictions and uses can be given to the application, as well as which data set and its use will require the application, allowing the end user to give the approval for such use.

But this development, like any other, has led to the generation of an impact on the environment in which it is defined and that is given by a set of factors.

During the last century and due to great technological advances, human health has been directly impacted, given the facilities and comforts that have generated in society. The fact of using the car or the bus, instead of walking, or shopping through the web instead of going to supermarket, are habits, which directly influence on the progressive sedentary lifestyle we are suffering from.

This leads an increase in diseases related directly to overweight or obesity, which includes diabetes. But this problem is also clearly related to the level of development of the countries, as there is a much higher rate of population suffering from such diseases. Another major cause also linked to this type of disease, is based on the increasingly poor eating habits that our society is taking, at a potentially dangerous pace. Fast-food restaurants, over-processed foods, food industries and another factors which are creating a society at risk that should no doubt be stopped as soon as possible.

But there is another aspect that place a major importance in the development of an application of these characteristics, this aspect focuses on the concept of **Digital Health**. This initiative emerges through the concepts of linking the great levels of digitization and technological development with the health services that, as explained, are increasingly necessary in today's society, not only for the prevention and treatment of chronic diseases, if not by factors such as aging and clinical needs of society, cost recovery and privatization of the sector, etc. As a result of this set of factors, this initiative is born, which aims to establish a global community, as wide as possible, specifically dedicated to this much needed and demanded sector. It should be noted that the initiative was born through a multitude of professionals in the sector, favoring in turn the growth of research in all areas, generating new possibilities of control, progress, information and innovation.

This initiative is one of the points of interest for the creation of our system, which seeks above all other needs, to promote growth and awareness about the disease that people with diabetes are, therefore, is a great contribution as to its impact on society, since the project can rely on an immense professional community behind it.

In this way, the concept mentioned above would reveal the ease of a need that every patient demands, such as public health and the right to possess through taxation to optimal clinical needs. This concept, implemented in a large number of countries, is not so common in non-developed countries or where society is more vulnerable to diseases of this type. For this reason, our system intends to enter fully and offer a new set of possibilities for patients, limiting the number of visits or reviews that, for example, a person may need to perform, for the correct control of their illness, or through intercommunication with doctors or specialized centers, you favor the knowledge that social groups or types of patients may be more likely

to need such services, thus saving a lot of costs, and offering patients who really need a better service.

But technological progress could not only help patients and governments save costs in terms of public health, but also open a new field for the distribution and management of medicines and accessories needed to control these diseases. For a diabetic person, the highest percentage of costs for proper maintenance of their health comes from the need for clinical materials, such as test strips, measuring devices, or injectable insulin.

This material is not characterized by its low cost, being around 20-25 euros a simple insulin pen, so that the distribution and its regulation can be critical for many patients.

So, after analyzing this environment in which our designed system is developed, we must add that the characteristics offered by our application are only a first step to the generation of a common idea or project, to be able to establish a development community on the world of diabetes.

Therefore, it should be noted that our product will be treated as free code software, re-usable and modifiable through electronic means and personal development, and that will try to evolve and develop, to the greatest extent possible, with the support of The health community, complementing, and applying new solutions to problems that may arise during the evolution of this disease. In this way mutual enrichment to advance a project capable of encompassing such a large group

Conclusions

The implementation of the "**GDMSCCT: GLUCOSE DATA MONITORING SYSTEM AND DIABETES CONTROL TIPS**" system has been created due to some needs in the sector. Firstly because of the current society and its way of life which mainly includes an unbalanced diet and sedentary jobs. Secondly it also may help the life of all those who are not in reach countries. Finally, it will contribute to implement technology in every aspect of our daily life, making it much easier.

In this way, the implementation and generation of this idea will allow patients suffering from Type 1 Diabetes, monitoring glycemic control and also being able to obtain better results in their tests. Lastly it will improve their standard of living through health.

With regard to the development of the project, an attempt has been made to define an adequate development for the type of audience to which it is focused, and giving the greatest number of people the opportunity to use the system. So it was decided to deploy on Android as the most widely used and globally modifiable operating system.

The greatest invest time was used in the creation of an optimal, consistent and solid system design, which will develop the greatest number of necessary functionalities possible from the point of view of the end user. This investment of time, has provided an adequate and capable system, which offers the user another point of view and abstraction towards his illness, and allows a better control of this, by offering relevant and useful data for correction.

The deadlines that were raised in the project have been fulfilled, being a key point of the development and implementation of the same, since the combination with other works and projects of the university have been present during much of this development.

As the last point of this development, I would like to call for awareness of health in our lives, because enjoying the possibilities of growth in all other aspects that encompass society would be much better developed. But in the second stay I would like to call for awareness of the treatment of Diabetes, whose gravity may not have importance in the mentality of citizens, and acts as simple as using means of transportation such as cycling, walking or avoiding extra processed food, could avoid suffering from this disease.

According to future applications that can be derived from the one we are talking about, GDMSCCT aims to establish a new vision, closely linked to the concept of **Digital Health**, which has already been discussed in this document. The intention is to provide a first step towards the creation of a community of analysis, experimentation and awareness, to treat Diabetes disease.

The fact that the system is Open Source or free code, supports development, and will at any time download any source code repository, edit, modify and release new versions, improved, or simply offer new capabilities and advantages to The life of patients who wish to use it. In this way, this software will have a virtually infinite capacity to grow and support more users, and doing it in a better way.

On the other hand, the system will support the analysis and the experimentation, since the rise of the new technologies of massive analysis or Big Data, could generate in the future an exceptional database on data captured by this application, being able to generate algorithms of prediction or of Deep Learning, able to help patients infinitely more accurately.

In addition to all this, the future of the application is very promising at the user level, as this may in the future, simply use your device to record the data, in an easy and simple, avoiding that intermediary companies must submit patients to use their own devices, raising the cost and profit at their expenses.

11- Referencias

- [1] A.D.A.M. Editorial team (Traducción y localización realizada por: DrTango, Inc.), 2015. [En línea]. Available: <https://medlineplus.gov/spanish/ency/article/000305.htm>.
- [2] C. A. Diaz, «Enter.co,» 2011. [En línea]. Available: <http://www.enter.co/cultura-digital/negocios/casi-el-75-de-la-poblacion-mundial-tiene-un-telefono-movil/>.
- [3] O. L. Londoño Palacio, L. F. Maldonado Granados y L. C. Calderón Villafañez, 2014. [En línea]. Available: http://www.colombiaaprende.edu.co/html/investigadores/1609/articles-322806_recurso_1.pdf.
- [4] International Diabetes Federation, [En línea]. Available: <http://www.idf.org/about-diabetes/what-is-diabetes.html>.
- [5] IDF Diabetes Atlas, 2015. [En línea]. Available: <http://www.diabetesatlas.org/>.
- [6] OMS, «Diabetes según la OMS,» 2016. [En línea]. Available: <http://www.who.int/mediacentre/factsheets/fs312/es/>.
- [7] J. Serra Grima, Corazón y ejercicio físico en la infancia y adolescencia, Barcelona: Masson, 2001.
- [8] A. Francisco Javier, A. Gutiérrez, Á. Caballero, J. E. Campillo, S. Murillo y A. Pérez , Diabetes y Ejercicio, Madrid: Mayo Ediciones S.A., 2006.
- [9] ITU World Telecommunication/ICT Indicators Database, 2017. [En línea]. Available: <http://www.itu.int/es/Pages/default.aspx>.
- [10] Xataka Android, 2011. [En línea]. Available: <https://www.xatakandroid.com/sistema-operativo/que-es-android>.
- [11] Open Handset Alliance, [En línea]. Available: <https://www.openhandsetalliance.com/>.
- [12] TopAppCreators, 2016. [En línea]. Available: <http://www.topappcreators.com/lip-smacking-evolution-android/>.
- [13] Android OS, 2012. [En línea]. Available: <http://androidos.readthedocs.io/en/latest/data/historia/>.
- [14] Android Security, 2017. [En línea]. Available: <https://source.android.com/security/>.
- [15] Android, 2016. [En línea]. Available: <https://www.android.com/>.
- [16] Abbot Diabetes, [En línea]. Available: <http://www.abbottdiabetescare.es/freestyle-insulinx>.
- [17] NFC.org, 2017. [En línea]. Available: <http://nearfieldcommunication.org/>.

- [18] SCIENCE DIRECT, 2016. [En línea]. Available: <http://www.sciencedirect.com/science/article/pii/S0716864016300116>.
- [19] GooglePlay, «Google Play Store,» [En línea]. Available: <https://play.google.com/store/apps/details?id=com.socialdiabetes.android&hl=es>.
- [20] Google Inc., «Material Design Guideline,» 2017. [En línea]. Available: <https://material.io/guidelines/>.
- [21] GooglePlay, «Google Play Store,» [En línea]. Available: <https://play.google.com/store/apps/details?id=com.mysugr.android.companion&hl=es>.
- [22] GooglePlay, «GooglePlay,» [En línea]. Available: <https://play.google.com/store/apps/details?id=io.huannguyen.diabetesmonitor&hl=es>.
- [23] LibreView Abbot Diabetes, [En línea]. Available: <https://www1.libreview.com/Dashboard/UploadMeter?newPracticePrompt=False>.
- [24] R. S. Pressman, Ingeniería del software. Un enfoque práctico, McGrawHill, 2010.
- [25] IEEE Std 830-1998, «IEEE Recommended Practice for Software Requirements Specifications,» 1998.
- [26] A. Durán Toro y B. Bernárdez Jiménez, «Metodología para la Elicitación de Requisitos de Sistemas Software,» 2000.
- [27] G. Booch y otros, Objet-Oriented Analysis and Design with Applications, Addison-Wesley, 2004.
- [28] M. Fowler, UML Distilled: A Brief Guide to the Standard Object Modeling Lenguaje, Addison-Wesley, 2004.
- [29] w. o. UML.org, «www.uml.org,» [En línea].
- [30] P. Kruchten, «Architectual Blueprints - the "4+1" View Model of Software Architecture,» 1995.
- [31] Android-Developers, «Android-Studio Blog,» [En línea]. Available: <https://developer.android.com/studio/projects/create-project.html>.
- [32] Android Developer, 2017. [En línea]. Available: <https://developer.android.com/studio/features.html?hl=es-419>.
- [33] F. Ableson, R. Sen y C. King, Android in Action, Manning Publications, 2011.
- [34] Android-Developers, «Android Developers Blog,» [En línea]. Available: <https://android-developers.googleblog.com/2016/09/android-studio-2-2.html>.
- [35] P. Jay, «GitHub Phil Jay,» [En línea]. Available: <https://github.com/PhilJay/MPAndroidChart>.

- [36] *Ley 41/2002, de 14 de noviembre, básica reguladora de la autonomía del paciente y de derechos y obligaciones en materia de información y documentación clínica.*, 2002.
- [37] *Directiva 2002/58/CE del Parlamento Europeo y del Consejo, de 12 de julio de 2002, Tratamiento de los datos personales y a la protección de la intimidad en el sector de las comunicaciones electrónicas*, 2002.
- [38] *Ley Orgánica 15/1999, de 13 de diciembre, Ley de Protección de Datos de Carácter Personal.*, 1999.
- [39] *Protection of individuals with regard to the processing of personal data and on the free movement of such data*, 1995.
- [40] OMS, «Obesidad y sobrepeso según la OMS,» 2016. [En línea]. Available: <http://www.who.int/mediacentre/factsheets/fs311/es/>.
- [41] Organización de la Salud Digital, «Salud-digital.es,» 2016. [En línea]. Available: <http://salud-digital.es/>.
- [42] WashingtonPost, «The Washington Post,» [En línea]. Available: https://www.washingtonpost.com/news/wonk/wp/2016/10/31/why-insulin-prices-have-kept-rising-for-95-years/?utm_term=.6364921d3611.
- [43] CDC, «Centers for Disease Control and Prevention,» [En línea]. Available: <https://www.cdc.gov/diabetes/statistics/prev/national/figpersons.htm>.
- [44] W. H. O. «Global Status Report on noncommunicable diseases,» 2014.